DID YOU KNOW?

We still have the very best PD library for the Atari Classic

There are over 400 disks available

Many disks are only £1.50 each

Collections and special sets are even cheaper

EVERY PURCHASE OF A PD DISK HELPS TO SUPPORT THE MAGAZINE

DID YOU REALISE?

We still have the entire ST library available

There are over 1,000 disks

Disks are only £1 each

We will send you details on request

KEEP SUPPORTING US AND WE WILL KEEP SUPPORTING YOU

Page 6 Publishing's

NEW ATARI USER

The Resource for the ATARI CLASSIC and the ATARI ST

Issue 85 - Autumn 1998 £2.50

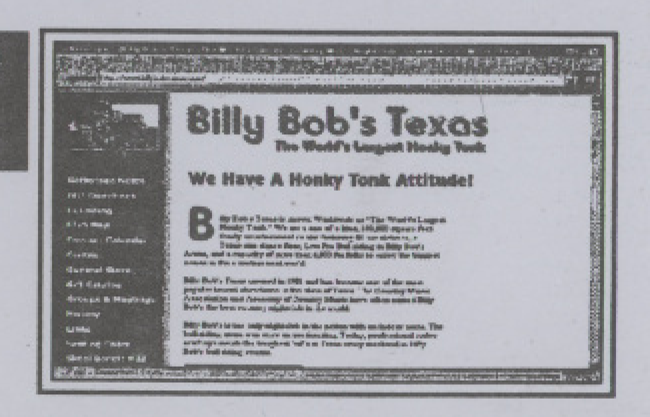
FOR THE ATARI CLASSIC

GENUINE PERUVIAN GUANO

- ***LEAGUE TABLE***
  Follow your favourite teams
- ***SUPER FILE READER***
  A neat simple way to read doc files
- **A SHORT HISTORY OF COMPUTERS**
  Celebrate the 50th anniversary of the computer

THE WIDER SCENE

HOW TO PLAN YOUR VACATION ON THE INTERNET



PLUS ... SUPERCIRCLES ... AUTOMATIC PROGRAMMING ... TURBO TUTORIALS ... THE TIPSTER ... AND MORE!

This issue's ....

Thanks

Les Ellingham puts it all together and fills up the gaps but the real thanks goes to the following who made this issue possible

Sandy Ellingham who takes care of all the office work, advertising and mail order

For their contributions this issue

James Austin
W H J Porter
John Tanner
John Robinson
John Harrington
Allan Palmer
Joel Goodwin
John Foskett
H S Wood
Austin Hillman
James Matthrick
Gavan Moran
John S Davison

APOLOGIES

I am still extremely poor in acknowledging contributions so I apologise to everyone who has sent in stuff and thought it has gone through the wormhole. The intention to reply to everyone is there but the time seems to drift by. If you have not heard, thank you and keep watching the mag, you might be surprised.

HOW IT'S DONE

PAGE 6 shows just what you can do with your Atari. NEW ATARI USER has always been created entirely with Atari equipment, initially on the XL but more lately with a Mega ST and other stuff, who needs PC's or Macs! Hardware includes a Mega ST2 (upgraded to 4Mb), SM125 Monitor, Supra 30Mb Hard Disk, a HP Laserjet III, Citizen 124D printer, Philips CM8833 monitor, 130XE, a couple of 1050 disk drives, 850 interface, NEC 8023 printer. Principal software used is Protext and Fleet Street Publisher 3.0. Other software includes Kermit, TariTalk, Turbo Basic and various custom written programs on the XL/XE. Articles submitted on XL/XE disks are transferred across to the ST via TARITALK. Programs are coded on the XE and printed out directly for pasting in after the typesetting is completed. All major editing is done with Protext and pages are laid out with Fleet Street Publisher. Each page is output directly from Fleet Street to a HP Laserjet III which produces finished pages exactly as you see them. All that is left is to drop in the listings and photos.

Well, it's not quite as easy as that but you get the idea!

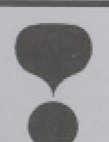
Inspiration

I first started this issue several months ago so I haven't got a clue what I was listening to at that time. As I type this Celine Dion is playing but that is a little unusual these days as most of my listening has been of Native American Music. Nothing new - I can't afford it - only things that I have mentioned in past issues. Oh, I almost forgot - Ann O'Driscoll very kindly sent me a CD called A Woman's Heart II and pretty good it is too with a couple of tracks that were already favourites from Dolores Keane and Maura O'Connell. Your kindness is very much appreciated, Ann, in what is becoming a gloomier world.

CONTRIBUTIONS

Without contributions from its readers, NEW ATARI USER would not be possible. PAGE 6 welcomes and encourages its readers to submit, articles, programs and reviews for publication. Programs must be submitted on disk or cassette, articles should wherever possible be submitted as text files on disk. We seek to encourage your participation and do not have strict rules for submissions. If something interests you, write a program or article and submit it!

COPYRIGHT



Editorial address: P.O. Box 54, Stafford, ST16 1DR, ENGLAND Tel. 01785 241153
Editor & Publisher: Les Ellingham - *Advertising:* Sandy Ellingham
Page layout by PAGE 6 - Printed by Dolphin Press, Fife, Scotland 01592 771652
NEW ATARI USER is published bi-monthly on the last Thursday of the month prior to cover date

PAGE 6 PUBLISHING's NEW ATARI USER

'The Magazine for the Dedicated Atari User'

ISSN No. 0958-7705

CONTENTS

Issue 85 - Autumn 1998

REGULARS

STILL A WHILE TO THE NEXT COPY DATE BUT PLEASE CONTRIBUTE NOW!

PROGRAMMING

FEATURES

COPY DEADLINE FOR NEXT ISSUE - 5th JANUARY

SUBSCRIPTIONS

MAGAZINE ONLY

Annual subscription rates (6 issues)

UK	£15.00
Europe (Air Mail)	£17.00
Elsewhere (Surface)	£17.00
Elsewhere (Air Mail)	£23.00

Overseas rates reflect only the difference in postal costs

DISK SUBSCRIPTION

A disk containing all of the 8-bit programs from each issue of NEW ATARI USER is available either separately or on subscription. Single price £2.95 per disk, a disk subscription saves you almost £8 a year. Subscription rates (6 issues)

UK	£25.00
Europe	£32.00
Elsewhere (sea)	£32.00
Elsewhere (Air)	£42.00

Please make cheques payable to PAGE 6 PUBLISHING and send to PAGE 6 Publishing, P.O. Box 54, Stafford, ST16 1DR

Editorial

THE FINAL COUNTDOWN

At last, another issue of NAU pops through your door. Thank you for keeping the faith and not hassling me too much.

Sadly all the good intentions for this year have gone to pot as the current government continues the previous government's policy of trying to destroy those who don't have conventional jobs. All this business about controlling inflation merely sacrifices people like me on the altar of those who already have most of the money. Increases in interest rates, and thus mortgage rates, stops people spending money which makes it more and more impossible for people like me to make a living. As you know I set aside a number of weekends this year to commit to NAU but because each craft fair that I have done this year has been worse than last year I have had to use each of those weekends to do more fairs in order to be able to stay afloat. That means, of course, no time to complete the mag. This month (October) has been so bad that it is almost certain that I will now have to find a conventional part-time job working nights, or whatever, in addition to the craft work. That of course means even less time. God knows how I will cope.

It seems obvious that I need to give you, the faithful supporter of NAU, a commitment that you will get the magazines you expect and whilst I cannot guarantee each issue will be on time I can guarantee that there will a certain number of future issues. I have therefore decided to begin the final countdown for Page 6/New Atari User. There will be six more issues of the mag after this one and then I will call it a day. It is unlikely that each issue will be on the bi-monthly deadline but you will get the mags you expect. We are already the longest published Atari magazine and my aim now is to be the first Atari magazine to close publication without its readers losing money on their subscriptions. As your subscription falls due you will be asked to renew only for the number of issues remaining.

As each issue is published it will become more and more difficult for me financially since there will be less and less renewal income. I therefore ask you to please ensure that you renew your subscription for the remaining issues, even if you no longer use your machine as much as you used to. Your commitment thus far has been much commendable, please now stick with us to the end. It is vital that we receive your continuing support.

Here's what you can do:

- *Renew your subscription when requested and know that you will get the rest of your mags*
- *Continue to buy a few PD disks now and then as this is our only means now of additional support*
- *Make sure that you contribute to the remaining issues by writing letters, articles or sending programs*

I am committed to giving you six more issues of NAU to take us up to the millennium, but I need your support to do that. Let's all stick together and bow out gracefully, knowing that we created something great over all these years.

Les Ellingham



PROGRAMMING

SUPER FILE READER

by James Austin

The Super File Reader (SFR) is a document file reader in Turbo-BASIC (XL/XE computers only) that I have written to enable quick and easy reading of document files in a properly presented fashion on-screen.

SFR has several features which set it apart from most other programs of its type. Firstly, it loads in as much of a document as it can in one go, so going some way to removing the need to keep the disk in when reading the file. Secondly, it automatically formats all documents during the reading process to cater for word-wrap and to ensure correct presentation when the files are displayed on-screen. And thirdly, it allows you to go back and forth at leisure through your documents, so you can go back and look at anything you might have missed or want to read again.

Although all references made in these instructions are to DOCUMENT files, it is also possible to load in and read other files if desired - however, it may be preferable to use a word processor program (such as TextPRO) for this purpose.

SFR has a text buffer of 26,500 bytes (approx. 26kB) which will allow document files of up to 212 sectors in size to be loaded in one pass - in practice this means that most document files will load in one go, however files larger than this can still be loaded in sections using the special oversize file routines built into the program.

I hope that you find SFR to be a useful program. Not exciting I know, but it was written to satisfy a need that I had and so it may well prove useful to you too.

THE MAIN MENU

All of SFR's operations are controlled through its main menu, which is displayed automatically at start-up. It displays 4 different options:

[1] Read in file
[2] Display file
[3] Disk Directory
[4] Exit SFR

To access these options, press [1], [2], [3] or [4] respectively. I will go through each in turn, starting with the main two options - [1] and [2].

Options [1] and [2] allow you to load in and display your document files respectively. A document file must have been loaded in previously and be resident in memory before option [2] can be accessed.

LOADING IN DOCUMENT FILES

Press [1] first of all. The main screen will clear, and the message 'READ WHICH FILE? :' is displayed. This is a prompt for you to enter the name of the document file you wish to load in, so enter it here and press [RETURN]. No device should be entered. Wildcards are not supported. Remember to have the proper disk in the drive before you hit the key, since the program immediately begins to scan the directory sectors on the disk to locate your file. If your file is not on the disk, then the prompt 'NO SUCH FILE ON DISK!!!' will appear, and you will be returned to the original prompt to try again.

If your file is found, then the program will immediately begin scanning that file's sectors. DO NOT TAKE THE DISK OUT AT ANY TIME DURING THE LOAD!!! - the program uses a machine code routine to access the Device Control Block and Internal Disk Handler routines and, although nothing nasty should happen, don't try it! You could crash the computer or even corrupt your disk in so doing.

If the prompt 'TOO LONG - FORMATTING PART LOADED...' appears, then the file is too long to load into the buffer in one pass. Bear this in mind for later on when I will explain how to load in oversize files. If the file successfully loads in one pass, then the prompt 'FORMATTING FILE...' will appear instead. The program is now formatting your document. The length of time this will take will depend on the length of the file you have loaded, also whether it is a true document or not. If you load a non-document file then you may well be waiting a LONG while as the program will have a bigger job on its hands than with a document file. Otherwise, you should not be kept waiting too long. Once this has been accomplished you will be returned to the main menu.

DISPLAYING DOCUMENT FILES

Now press [2]. The screen will clear and the first page of your document displayed on the screen. Notice the prompt 'PAGE x OF x' which appears at the bottom of the screen while this is being done - this tells you what page you are currently on, as well as how many pages there are in total. Once the page is completely shown on-screen, this is replaced by a small menu:

+ - BROWSE P CHOOSE PAGE ESC TO EXIT

Pressing [+] will display the next page in the sequence (if there is one), while [-] will move back to display the previous page. Pressing [ESC]ape will return you back to the main menu. If you press [P] then the screen clears and the message 'NEW PAGE #:' appears. This is a prompt to choose which page you wish to go directly to. Enter the page number of your choice and then press [RETURN]. Only valid page numbers will be accepted. The appropriate page will then be displayed and you will be returned to the above menu.

OVERSIZE FILES

Now on to the subject of reading in oversize files. To read the next part of the file, press [1] again - the message 'CONTINUE LOADING SAME FILE Y/N ?' will be displayed on-screen. If you press [Y] the next segment of the file will then be loaded in, formatted and displayed in the same way as described previously. If you press [N], then the load is terminated and the prompt to load an alternative file comes up. It should be noted that the previous part of the document file is lost when a new part is subsequently loaded in, so you will need to load the file from scratch to view previous pages again.

SFR will cope with as many additional parts of a document as are required to load the file in completely; it will also ensure that words are not broken between segments of a file but are instead presented in full on-screen when the next part is subsequently loaded in.

DISK DIRECTORY OPTION

Option [3] allows you to obtain a directory of the files on a disk in the default drive (usually drive 1). Ensure that a disk is in that drive before accessing this option.

The directory will be scanned and the files on the disk displayed in two columns on the screen. If there are too many files on the disk to display on the one screen, the message 'CONTINUE READING DIRECTORY Y/N ?' will appear at the bottom of the screen. Pressing [Y] at this point will clear the screen and display more of the directory, while pressing any other key will exit back to the main menu. If the files do all fit on the one screen then you will be asked simply to press a key to exit back to the main menu.

EXIT SFR

This is the final option available from the menu, accessed by pressing [4]. When this option is accessed, the prompt 'DO YOU WISH TO EXIT Y/N ?' will be displayed. Pressing [Y] will exit this program and place you in the Turbo-BASIC editor. Any other key will return to the main menu.

PLEASE NOTE...

SFR automatically inserts inverse-asterisks into your document file to format the pages so that they appear properly on-screen. This means that it is not possible to include inverse-asterisk characters in your documents, since this would corrupt the pagination and cause parts of your document to disappear. The program therefore converts all inverse-asterisks that it finds when loading the document into "normal" (non-inversed) ones.

A BUG IN TURBO-BASIC?

Now here is a question for all experienced readers out there. Is there a bug with the INSTR/UINSTR commands in Turbo-BASIC?

```
10 DIM TEXT$(3):TEXT$="*@*"
20 LET LOC=INSTR(TEXT$,"*",1)
```

I found with this example that LOC equals 3, not 1 as I would have expected. What seems to be happening is that the command is "missing" the first asterisk in the string, possibly because it is beginning the search from the SECOND character in the string rather than from the first one as I wanted? Change line 20 to:

```
20 LET LOC=INSTR(TEXT$,"*",0)
```

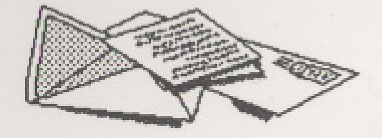
and LOC becomes 1, the answer that I thought the first example would have given me!

I wasted a LOT of time with this when writing SFR. Is there a bug?

FINALLY

If you experience any difficulties with using the program, don't hesitate to contact me at: *19 Clive Road, Bobbing, nr. Sittingbourne, Kent, ME10 1PJ, ENGLAND*, and I will try to help you out as best I can.

Mailbag



DID YOU WRITE?

Yet again we have one of those issues where only a few people decided to write in. I know the copy date for this issue was supposed to be really close to the date you received your last mag, but don't let that stop you writing. Write to us at any time, if your letter doesn't make it into the next issue it will go in the one after. We need your letters, they really help us out and let us know you are still interested.

Les Ellingham

WHICH PRINTER?

Reader W H J Porter from North Chingford has a problem that he would like your help with, which might also be a good starting point for an article:

"For many years I have used an Atari 1020 Printer/Plotter, but now it is not possible to obtain supplies and I would like to output text from disks such as Home Filing Manager. The problem is how do I modify the disk to output to my Epson printer?

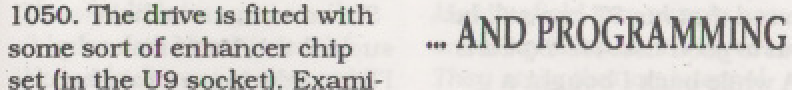
The problem also applies to other disks which I have purchased from Page 6 Library in past years. Perhaps one of your more expert readers may be able to answer this, or possibly write an article on the subject as I am sure that I am not the only reader with this problem"

❢ *I don't know about Home Filing Manager which probably has its own method of saving files but I can't see why you should have problems with Page 6 Library disks which have not be created for specific printers. Of course you may be talking about programs that were specifically written for the 1020. On ordinary DOS files you may like to try using DOS to dump the files to your Epson. All you do is call up DOS and use the Copy option (C) with the parameters filename.ext,P: which will dump the file filename.ext (of course use the name of the file you want to print) to your printer. You won't get any formatting but at least you'll get printed copy. There are many other ways of getting printed copy such as saving text versions of files to disk and then loading them into a word processor. Maybe this can be done with Home Filing Manager. Perhaps someone can check this out and write into Mailbag or put it together as an article.*

MORE 1050 PROBLEMS

John Tanner has a problem that the more practical hardware buffs can probably help with: "For many years, my family has enjoyed the use of our original purchase 800XL and the second 1050 which I acquired a few years ago.

The problem? A few weeks ago when the machine was taking a rest from the children, but was left powered up, I noticed a strange smell, then I saw a thin stream of smoke rising from the second



1050. The drive is fitted with some sort of enhancer chip set (in the U9 socket). Examination of the circuit revealed that two small capacitors (C65, C66?) immediately in front of large blue capacitor C68 and between TP14 and TP15 were completely burnt away, and had slightly damaged C68. I also noticed that CR15 and CR16 (2 diodes?) between the large capacitors C71 and C68 are both swollen and one is cracked.

Is it likely that the replacement of these components would restore the disk drive, or could some underlying fault have caused the overheating? Advice on repair would be most welcome as I have no service manual/information on the 1050.

If repair proves to be not a practicable solution a 'wanted' ad for a replacement drive would be the answer, as life with the single drive is so frustrating!

Incidentally both drives did read and write satisfactorily, although some disks seem to have a definite preference for one of them, however both give trouble with formatting with failures appearing to occur on the final tracks. Is there a simple reason for this?"

... AND PROGRAMMING

John now changes tack and fancies a bit of programming. His letter continues: " Now for something completely different. I would like to try my hand at programming our Atari (machine code that is), so I need some literature on the subject, including the memory layout of the machine. If there's someone out there who could help, I'd be eternally grateful.

Finally, a few years ago I began procedures to get a 3½" disk drive interface from Derek Fern, buy for various reasons at the time I let the matter drop. I now would like to purchase one of these interfaces if they are available."

❢ *Thanks for writing, John. I can't help with the 1050 problems (way above my head!) but I am sure that there are readers who can. As to the machine code writing you will have to get hold of some second-hand books as there is nothing currently in print. Best bet is for someone with some surplus books to get in touch with John direct. You can give him a ring on 01545 580625. Of course you might be able to help with the disk drive problem by talking to him but if you do please also write to Mailbag as I am sure that the solutions will be of interest to other readers.*

DIY BINDERS

John Robinson up in Cumbria has some more suggestions for making your own binders for NAU which he feels might be cheaper than Eddies Jones' method in Issue 84.

"Again, use the Rexel Budget binder number 13428 and obtain a pack of Grafix A4 punched pockets, usually sold in 10s but currently available from Asda and discount stationers for about £1 for 40. Take a pocket and fold it lengthwise, then cut in two with a very sharp knife. Discard the plain half. Fold the half with the white punched strip attached, and cut that in half again. The result will be two transparent pieces 152mm long by 115mm deep, each of which will hold one copy of NAU. Split the bottom of the pocket, to allow the two 'pages' to open out. Next, carefully mark the cover of your magazine 29mm from the top and 29mm from the bottom using a soft pencil. Obtain a UHU glue pen, and apply the gum to one inner side of the plastic (top, mid-



dle and bottom, then the outer edge in an E shape) and press the back of the magazine cover firmly onto this. Glue the other half, attach to the front cover and there you have it, a 'see through' means of utilising the spring clips in your binder.

Two tips - the second cut to halve the spine of the pocket will cut through a hole which is exactly in the centre of the punched spine - always ensure this half hole is at the same end of your mags, in the interests of neatness, and always use a very sharp knife to cut the pockets - they are tougher than you think and also very slippery!. Each binder holds six copies, very handily."

❢ *Many thanks, John, it is always helpful to share ideas. In closing John has asked if anyone has a copy of the SYNCALC spreadsheet they no longer want as he is desperate for a copy. If you send it in to us, we will pass it on.*

KEEP YOUR ST!

John Harrington is one of those people who 'upgraded' to a PC but who has now found that his ST can still be put to good use. He explains -

"A while back I bought a batch of ST PD stuff from you. My reasons were mainly selfish - to get my three boys to leave the PC alone so I could use it! My lads are all aged under 8 and the GEM interface is not as familiar to them as Windows 95 so initially they were all reluctant to use the ST, but on Sunday I finally found the time to set the ST up and go through the dozen or so disks you sent me. The inevitable result was that the three ended up fighting over who had command of the joysticks. In between punch-ups it kept them amused for about 6 hours which means I have already had my money's worth.

The two young 'uns (aged 4 and 6) particularly enjoyed the Mr. Potato Head and Spitting Fish games whilst the eldest enjoyed PacMan which, so far as I can see, is indistinguishable from the arcade version.

Okay, some of the software is incredibly rudimentary - the arithmetic tests on one disk are just text and random numbers - but for the infant school generation it (a) does not seem to matter too much and (b) it's helpful to them in their work.

So, as soon as we have sucked this lot of disks dry I'll be ordering another batch. The only drawback with the ST is the lack of a hard drive - it would be great if the files were just one (double) click away from running once the ST is switched on. So, although it defeats the object of encouraging the kids to use their own computer and not my PC, I downloaded an Atari ST emulator from http://pacifist.fataldesign.com/ this week. I can tell you it really was eerie seeing the GEM desktop on my PC. I'm not entirely sure I have got it up and running properly yet as I tried an old favourite game of mine called Gridiron and it said the disks were unreadable. I haven't had the time to try them out on the ST yet to see whether the disks are merely unreadable on the PC's disk drive or genuinely damaged.

There are, of course, emulators for the Atari Classic and I downloaded the files for that too. The drawback is that although I have a 5¼" disk drive on my PC, it won't read Atari floppies. I've not read the documentation thoroughly yet but it appears there is some cabling needed to hook up my Atari disk drive to the PC. There are some Atari Classic 'disk images' available for download from the Web, which means you can run those directly from your hard drive, but most of the games I want to run are strategy games (Rails West!, M*U*L*E*, Seven Cities of Gold) and not the arcade ditties that most people seem to run of their Atari.

I know your correspondents have covered emulators in previous issues but it might be an idea if those people still supporting the Atari Classic were to put together an 'emulator package' for lazy thickos like me and publicize it through your magazine."

❢ *Bet you are now glad that you didn't sell your ST for a pittance at a car boot sale like so many other people! The obvious answer to the lack of a hard drive for your ST is to buy one! Believe it or not you can still purchase a hard drive for the ST (at least you could a couple of months ago). They are not quite as cheap as those available for the PC but are considerably less that the £300/£400 you had to pay for the Supra drives. Because the ST is still widely used in music environments there is quite a demand for upgrades and one of the music mags recently reported that The Upgrade Shop in Macclesfield can supply some amazing hardware upgrades. They state that an internal hard disk drive can be added to an STE for £100 for a 170Mb model going up to £160 for 720Mb. You can fit it yourself or have it fitted for you for £15 plus £7 courier collection and delivery in each direction. A couple of years ago The Upgrade Shop were also advertising external hard drives for the ST (at a much higher price, although this is sure to have come down) and these may still be available. If you are used to using a PC hard drive then you might think that 170Mb is a rather small drive but the PC is incredibly inefficient in data storage and 170Mb on the ST is more than you are ever likely to need. I have a 20Mb hard drive and it copes adequately with all I have to do. Just think, with 170Mb you can store something like 230 single sided ST disks!*

If you want to get more details give The Upgrade Shop a ring on 01625 503448.

RANDOM NOTES

Allan Palmer has promised to organise some of his ramblings into an article for us (committed you now, Allan!) but in the meantime has a few random notes of interest:

"Any information on Atari Simulators for PCs would be more than welcome; hopefully some of our readers can elaborate more.

- Kevin Cooke's letter in the last issue had plenty of interesting comments, although as you'd probably guess from my last letter, I disagree with Kevin that the Internet/World Wide Web has given him limited success. You need to put a bit of effort into a search for what you need and be specific, "Sick Building Syndrome" would have been a better search text than just "SBS" ...

Finally could you please note my new e-mail address which is now Allan_Palmer@bigfoot.com"

❢ *Many thanks Allan, your e-mail address should now be correct in our Internet Contacts column. Your point about searching on the Internet is well taken although I continue to have the odd dabble and I still can't get to grips with refining searches. The problem is that all of the search engines seem to use different criteria for their search with some requiring inverted commas around a*





phrase, others needing underlining between linked words and yet others needing plus signs on words to be included. It is okay if you have unlimited access to the Internet at home or at work (even better if you don't pay the phone bill!) but when you have a hour at the library, it is difficult to learn all the different ways of using various programs. Perhaps, Allan, you or some other reader could do us a short article on the best way to use various search engines, using some sort of practical example? I know that there are various books around that may give you this information but it would be handy to have it in one place. Incidentally Staffordshire Libraries are now hooked up to the Internet and very reasonably priced at just £1.50 per half hour with printouts at just 10p per copy. The main trouble is that no-one in the library has the remotest idea of how to use the World Wide Web and can't give a novice any help whatsoever. It astonishes me that you have to have all sorts of degrees to become a librarian yet when it comes to learning about what will be the future of information gathering and research, none of them seem to want to know!

GOT A PC?

Here's a way that you might be able to help other readers. We have to face up to the fact that a lot of our readers will eventually go over to using a PC, even if they intend to keep their Atari systems. Specifications of PCs change almost weekly so it is virtually impossible to recommend a particular set-up as being the best for a particular purpose. If you already have a PC, how about writing to us to let us know what it is. Who makes it? What processor do you have? What 'extras' are installed? Has it been reliable? How often does it break down? Does the manufacturer give good service? Does it do all you wanted it to do? Would you now go for something different?

Tell us about your experiences for the next Mailbag, or for a special article, and your experience and advice could help others in their decisions. Reading reviews in PC magazines is pretty useless, as all they want to do is push the latest supersystem and they make it sound as if you may as well not bother unless you have the latest 450Mhz processor with a DVD drive, and to them £1,000 is peanuts!

Your practical experience is far more valuable to readers who want to make the commitment, so share it with us.

And .. believe it or not folks, that was it for Mailbag this issue even though there has been a long gap between issues. When I first started this issue a couple of months ago we only had two letters! Let's have a lot more for the next one, there is still plenty to talk about. The address, as always, is:

MAILBAG
NEW ATARI USER
P.O. BOX 54
STAFFORD
ST16 1DR



PROGRAMMING

SUPERCIRCLES

Joel Goodwin does the maths to help you to draw the perfect circle - well, almost!

Drawing a circle is something that is necessary from time to time but the standard routine is slow. This article describes an attempt to speed up the routine by using an alternative approach.

THE IDEA

The usual way of drawing a circle is to use Pythagoras' theorem. You can use a SIN/COS function approach, but this turns out to be slower. Pythagoras' theorem relates the radius to the X and Y coordinates of a point on the circle, so by varying either the X or Y coordinate we can determine the other. However, the formula used is Y=SQR(R*R-X*X) which involves a square root function; this is what drags the subroutine down. We can optimise the method by using the circle's symmetry to reduce the amount of work we need to do, but SQR is slow and ideally we would like to replace it with something more efficient.

The SQR-routine treats the circle as a collection of points and each point is found individually. Another approach is to try to follow the path of the circle; that is, each new point is thought of as an "update" of the previous point. This is possible through an approximation of the slope of the circle; it involves a little calculus, so if you're interested see the Mathematics section at the end of this article. This new approach replaces the SQR function by an addition and can also be accelerated by relying on the symmetry of the circle. The resulting "Supercircle" subroutine is demonstrated in Listing 1.

THE DEMONSTRATION

The demonstration program will first compare the speed of the standard and supercircle routines. After this, it will then attempt to erase standard circles with supercircles. This will show you that the supercircles are not the same as the standard circles; personally, I prefer the supercircles but it is up to you to decide which one you like best. The program will then show you a filled supercircle and finally proceed to draw two large circles in graphics mode 8 (one is a standard circle, the other is a supercircle).

The difference in speed will not be so signi-

```
NP 2 REM ==============================
XD 3 REM =    SUPERCIRCLES DEMO      =
PK 4 REM =       Joel Goodwin        =
JP 5 REM =  NEW ATARI USER - 1998    =
NT 6 REM ==============================
XS 20 GRAPHICS 7:POKE 710,4:POKE 708,184:
   POKE 709,26:POKE 752,1:DIM M$(40)
EN 30 M$="First we draw 2 normal circles.
   ":GOSUB 4000
FX 40 COLOR 1:XC=20:YC=20:R=19:GOSUB 1000
AK 50 XC=139:YC=59:GOSUB 1000
MF 60 M$="Now compare with 2 supercircles
   .":GOSUB 4000
CA 70 COLOR 2:XC=139:YC=20:GOSUB 2000
SN 80 XC=20:YC=59:GOSUB 2000
WM 90 M$="Watch supercircles erase normal
    ones.":GOSUB 4000
XG 100 XC=79:YC=39
DQ 110 FOR R=39 TO 4 STEP -5
DY 120 COLOR 1:GOSUB 1000
EI 130 COLOR 0:GOSUB 2000
JL 140 NEXT R
CP 150 M$="We can also fill a large super
   circle.":GOSUB 4000
UU 160 XC=79:YC=39:COLOR 2:R=39:GOSUB 300
   0
TI 170 M$="Finally two huge mode 8 circle
   s.":GOSUB 4000
CD 180 GRAPHICS 24:POKE 709,12:POKE 710,0
PD 190 COLOR 1:XC=95:YC=95:R=95:GOSUB 100
   0
VB 200 XC=224:GOSUB 2000
ME 210 GOTO 210
TN 999 REM  NORMAL CIRCLE ROUTINE
FX 1000 FOR DX=0 TO R/SQR(2)
FA 1010 DY=SQR(R*R-DX*DX)
MQ 1020 PLOT XC+DX,YC+DY:PLOT XC-DX,YC+DY
YC 1025 REM DX=INT(R*SIN(THETA)):DY=INT(R
   *COS(THETA))
QZ 1030 PLOT XC+DX,YC-DY:PLOT XC-DX,YC-DY
MK 1040 PLOT XC+DY,YC+DX:PLOT XC-DY,YC+DX
QT 1050 PLOT XC+DY,YC-DX:PLOT XC-DY,YC-DX
NU 1060 NEXT DX:RETURN
XS 1999 REM  SUPERCIRCLE ROUTINE
GH 2000 DX=0:DY=R
MO 2010 PLOT XC+DX,YC+DY:PLOT XC-DX,YC+DY
QX 2020 PLOT XC+DX,YC-DY:PLOT XC-DX,YC-DY
MI 2030 PLOT XC+DY,YC+DX:PLOT XC-DY,YC+DX
QR 2040 PLOT XC+DY,YC-DX:PLOT XC-DY,YC-DX
HA 2050 DY=DY-(DX/DY):DX=DX+1
ID 2055 REM DX=DX+1:DY=DY-(DX/DY)
XG 2060 IF DX<=DY+1 THEN 2010
AV 2070 RETURN
JV 2999 REM  FILLED SUPERCIRCLE ROUTINE
GI 3000 DX=0:DY=R
VZ 3010 PLOT XC+DX,YC+DY:DRAWTO XC-DX,YC+
   DY
AM 3020 PLOT XC+DX,YC-DY:DRAWTO XC-DX,YC-
   DY
VT 3030 PLOT XC+DY,YC+DX:DRAWTO XC-DY,YC+
   DX
AG 3040 PLOT XC+DY,YC-DX:DRAWTO XC-DY,YC-
   DX
HB 3050 DY=DY-(DX/DY):DX=DX+1
YE 3060 IF DX<=DY+1 THEN 3010
AW 3070 RETURN
IW 3999 REM  PROMPT USER FOR KEYPRESS
IV 4000 ? CHR$(125):POKE 657,20-LEN(M$)/2
   :POKE 656,1:? M$
UK 4010 ? "      Press any key to continue
   _";
GA 4020 POKE 764,255
VR 4030 IF PEEK(764)=255 THEN 4030
SJ 4040 ? "[ESC,DELETE LINE]";:POKE 764,2
   55:RETURN
```

Underline = INVERSE CHARACTERS · [ ] = CONTROL + CHARACTER · < > = INVERSE CONTROL + CHARACTER

ficant if the program is run under Turbo Basic, but it is still noticeable. Turbo Basic has its own CIRCLE command, which is very fast (it is written in ML, after all) and produces a circle which looks different to the subroutines presented here.

The program contains three subroutines you may find of interest:

1000 - Draw standard circle
2000 - Draw supercircle
3000 - Draw filled supercircle

Each requires XC and YC to be the coordinates of the centre of the circle and R to be the radius, but they do not alter these variables. You should also use a COLOR command to select the colour of the circle prior to the "GOSUB <circle subroutine>". (Remember that placing a routine at the start of a program may improve its speed.)

The supercircle routine can generate a different circle by swapping the order of the updates to the X and Y coordinates; the commands after the REM on line 2055 can be used to replace those on line 2050 if you wish to see the effect. Note that deleting 2055 altogether will make the subroutine a little more efficient, as the line is superfluous to the routine.

THE BUG?

You may notice that the large mode 8 standard circle seems to suffer from a little wiggle at the top and left edges of the screen. This is not a bug in the method but actually in Basic itself.

Listing 2 is a program which isolates the bug. What this program does is to plot the points corresponding to the following sequence of coordinates:

(0,0.5)	(0,10.5)
(1,0.05)	(1,10.05)
(2,0.005)	(2,10.005)
(3,0.0005)	(3,10.0005)
...	...

You would expect all of the points to form two straight lines except for the first pair of points which should be just below the line (Basic should round up 0.5 to 1 and 10.5 to 11). This prediction is correct for the second line, but the first wobbles up and down!

I have no explanation for this phenomenon, but it doesn't happen if you replace 0.5 with a smaller number like 0.1, 0.4 or even 0.4999. The wobble is clearly connected with Basic's rounding-up mechanism - this bug is NOT present in Turbo Basic. The fact that the supercircle subroutine does not exhibit this problem demonstrates further how robust the new method is.

THE END

I have not seen this subroutine developed elsewhere, so I am assuming it is new. The supercircle method could be applied without too much difficulty in machine language, although floating point numbers would be

continued on page 46

```
SY 10 REM  SUPERCIRCLES BASIC BUG?
KW 11 REM       Joel Goodwin
WD 20 GRAPHICS 3:COLOR 1
WO 30 Y=0.5
RC 40 FOR X=0 TO 39
CE 50 PLOT X,Y:PLOT X,Y+10
VU 60 Y=Y/10
OA 70 NEXT X
```



PROGRAMMING

LEAGUE TABLE

by John Foskett

When organising sporting events where many teams compete against each other, some form of correlating the individual scores and displaying the overall results in a clear and concise way is required. The organisation of our national game over many years has lead to the football league tables as we know them today and "League Table" was written to reflect the same principles exactly. In the same way as the Nationwide football league and the Carling Premiership, League Table gives 3 points for a win, 1 point for a draw and no points at all for a loss. League Table allows you to create new blank league tables, to update them by entering the individual scores and to display the tables on screen together with the option to print the tables out using an Epson compatible printer. League Table also sorts the competing teams into order in accordance with the points accumulated and where teams are on equal points, using the goal difference in the same way as the football league tables.

THE MENU

Upon running the program, the following menu is presented and each option on the menu is described below....

(A) LOAD DATA FROM DISK
(B) SAVE DATA TO DISK
(C) SORT DATA
(D) CREATE LEAGUE TABLE
(E) DISPLAY LEAGUE TABLE
(F) ENTER RESULTS (UPDATE)
(G) LIST FILE DIRECTORY

(A) LOAD DATA FROM DISK: Upon selecting this option, a number from 0 to 9 is prompted for to access one of the ten data files previously saved to the disk. If the selected file in not on the disk, then an error will result and ESCAPE must be pressed to exit. ESCAPE may also be pressed to exit from the prompt without loading data and without corrupting the data currently in memory.

(B) SAVE DATA TO DISK: When this option is selected, a number from 0 to 9 is prompted for in the same way as option 'A' above in order to access one of the ten data files. Selecting the number of a previously saved file will cause that file to be overwritten with the current data. If the selected file has previously been locked (using DOS), then an error will result requiring ESCAPE to be pressed to exit. ESCAPE may also be pressed from the prompt to exit without the current data being saved.

(C) SORT DATA: Immediately this option is selected, the current data in memory is sorted into order according to the number of points the teams have accumulated and if some teams are on the same number of points, then the goal differences are taken into account. The goal difference being the difference between the total number of goals a team has scored in the league and the total number of goals scored against them. This option can only be selected with data in memory, but if selected with no data in memory, then ESCAPE must be pressed to exit.

(D) CREATE LEAGUE TABLE: This option is used to create a new blank league table. If this option is selected with data currently in memory, then the option to continue is given by pressing either 'Y' for Yes or 'N' for No. If 'Y' is pressed to continue, then any data currently in memory is erased, but if 'N' for No is pressed, then any data currently in memory is preserved. A league may consist of up to a maximum of 24 teams or a minimum of 10 teams. The team names may consist of up to a maximum of 16 characters each in any combination of uppercase letters and spaces. Exit is automatic after entering the 24th team name, but TAB must be pressed to exit with less than 24 teams, but note that a minimum of 10 team names must be entered before TAB is enabled. After exiting in both cases, ESCAPE must be pressed to exit back to the menu. At this point the new blank league table data is contained in memory and so it is only a matter of saving it to disk using option 'B' as previously described. Note that pressing ESCAPE to exit at any other time will erase the newly created data from memory and also note that any team name entered which is a repeat of a previously entered team name either in full or in part will be rejected.

(E) DISPLAY LEAGUE TABLE: This option is used to display the data currently in memory on screen in the form of a league table, but note that for a true representation, the data in memory must first be sorted into order using option 'C' as previously stated. If this option is selected without any data in memory, then ESCAPE must be pressed to exit. League tables are displayed on screen in the same way as the football league tables are shown on television, with the team names in a column on the left followed by columns of figures on the right hand side. Following the team names are the number of games the teams have played, the number of points the teams have accumulated, the number of goals the teams have scored, the number of goals the teams have had scored against them and their overall goal difference. The league tables are displayed in four colour zones for clarity indicating the top or championship position in brown, the next three promotional positions in green, the relegation positions in red and the bulk of the table, the middle positions in the normal Atari blue. No matter how many teams are in the league, the bottom three positions will always be shown in red because the relegation zone is moved accordingly and if necessary the remaining part of the screen below the relegation zone will be blanked out with the background colour. ESCAPE is pressed to exit back to the menu or 'P' is pressed to printout the league table using an Epson compatible printer. Because of the extra width available when using a printer, 80 columns as opposed to 40 columns on screen, extra data has been included in the printout. When printed out, the league tables include the teams full home and away records, recording the total number of wins, draws and defeats for both home and away matches.

(F) ENTER RESULTS (UPDATE): This option is used to update a league table by entering the relevant results, but if this option is selected with no data in memory, then ESCAPE must be pressed to exit. All the team names in the league are displayed on screen in their correct league positions for reference, assuming of course that the data has previously been sorted into order using option 'C' as previously described. Entering the result (or the score) of a match is achieved by first entering the name of the 'home' team followed by the name of the 'away' team after which the result of the match is entered. The result is entered by pressing a numeral key (0 to 9), entering the home teams score first followed by the away teams score. When entering the team names, the names may be abbreviated to the first few letters if required, but be careful to match with the correct team, the full name still being printed on screen even if abbreviations are used. Normally only the first three or four letters need be entered, but it depends upon the names of the teams in the league and upon their league positions. To distinguish between Manchester Utd. and Manchester City will obviously require at least the first 12 letters to be entered. After entering the team names and the result, a prompt is given asking if the entered result is correct and if so 'Y' for Yes is pressed to enter the result or 'N' for No is pressed if the entry is in error. Upon pressing 'Y', the current data in memory is updated and the two teams are marked on screen accordingly with an inverse 'H' or an inverse 'A' to signify home or away for reference to show that the results for the marked teams have been entered. Note that these marks are erased when exiting and are not replaced should this option be reselected. If when prompted, 'N' was pressed, then the two team names and the result are cleared and the current data in memory is not updated. Entering the results continues in this way until ESCAPE is pressed to exit but note that option 'C' must be used to sort the data into order before a true league table can be displayed or printed out by using option 'E' as previously described.

(G) LIST FILE DIRECTORY: This option is used to list all the league table data files on the disk for reference. The program allows for up to ten files to be stored on disk and this option lists each one found as a single numeral each separated by commas after which ESCAPE must be pressed to exit.

USING THE PROGRAM

To clarify how to use the program, the first process before anything else can be achieved is to create a new blank league table using option 'D' and then save it to disk using option 'B'. Initially file 0 is best used to store a blank file. Load the blank file using option 'A', then use option 'F' to update it, use option 'C' to sort the data into order and then use option 'B' to save it back to disk, use option 'E' to display the league table on screen and if necessary to print out the league table.

Unlike most programs, League Table has no limits to the values that the data can accommodate and so theoretically the program can accept the normal six-byte values. Of course such large values will disrupt the screen display and the layout of the printout but it does have the advantage of keeping the league table data accurate. If all the results of a particular football league such as the Carling Premiership are entered into the program, then at the end of the season as well as during the season, the program will produce an accurate representation of the actual league table. The only discrepancy may be when two or more teams have the same number of points and the same goal difference, but as the season progresses, this becomes a very rare occurrence.

TECHNICAL DETAILS

ALTERING THE SIZE OF THE PROMOTION AND RELEGATION ZONES

The size of the programs promotion and relegation colour zones on screen are fixed, but they could be altered if required. The size of the promotional green zone is established directly by calling up the relevant DLI routine within the display list at the 52nd element of U$ on line 1860 which is an inverse CONTROL-B character.

To increase the size of the promotion zone by one position, move the character one position down the string to the 53rd position and replace the original 52nd character position with a normal CONTROL-B character. Alternatively to reduce the size of the promotion zone, Move the character up the string.

The size of the relegation red zone is established by a local POKE into the display list in page six within the DISPLAY procedure. The local POKE is referenced from the variable NUM which is used to store the number of teams in a particular league so that the relegation zone always covers the last three positions no matter how many teams are actually in the league. "POKE 1580+NUM,130" found on line 690 is used determine the size of the relegation zone and the corresponding "POKE 1580+NUM,%2" to cancel the DLI call can be found on line 750. Note that both must POKE the same address. Using "1579+NUM" will increase the size of the relegation zone whilst "1581+NUM" reduce the size.

If the size of the promotion and relegation zones are altered, then the position of the message "PRINTING TABLE PLEASE WAIT" displayed when downloading the data to a printer will have to be considered accordingly. The relevant three lines of the league table are temporarily MOVEd into U$ before they are overprinted on screen with the message after which they are MOVEd back into the screen RAM overwriting the message to restore the league table.

THE THREE DISPLAY LISTS

The first of the three display lists is used for the program's menu and employs mixed text modes. One line of mode two and three lines of mode one are used for displaying the programs titles and eight lines of mode zero for the menu.

The second of the display lists is used for displaying the league tables on screen and is a normal mode zero display list with three extra lines added at the top of the screen, one line of mode one for the programs title and two lines of mode zero for the league tables heading and its two option menu. This display list is located at the page six address of 1573. There are two asterisks at the top of the display list where the address of X$ (the extra lines screen RAM) is DPOKEd. X$ is defined on line 1890.

The third display list defined on line 1870 is a normal mode zero display list but lined and with an extra mode one line at the top of the screen for displaying the programs title. This display list is used for option 'D' to create new blank league tables and option 'F' to enter the results. This display list is located at the page six address of 1611 and it also contains two asterisks where the address of X$ is DPOKEd, but only the first 20 bytes of X$ are displayed since this display list only contains a single extra mode one line.

THE FIVE DLIs AND THE ZONE COLOURS

The program uses five DLI routines solely to provide the colours for the four colour zones used when displaying the league tables. The five DLI routines are defined together as U$ on line 1900 and MOVEd into page six following the display lists at address 1670.

The DLI routines are loaded with the relevant colour values using the unused locations 590 to 594 via the DATA statements at the end of the listing on line 1940. Since there

are only four colour zones in the league tables, only four DATA statements have been included on line 1940 which makes changing the DATA easier when altering the colours. The fifth colour value of 146, the background colour is POKEd directly into location 594 which is used to blank out the unused area of the screen below the relegation zone when a league contains less than the maximum of 24 teams.

THE VBI ROUTINE

A small immediate VBI routine is used to disable the attract mode, the CONTROL-1 stop-start toggle and to disable the lowercase and inverse characters ensuring that the keyboard is always in the uppercase mode. The VBI routine also resets the DLI vector register 512 to point to the first DLI routine to synchronise the screen colours with their respective zones when displaying league tables. The last DLI routine also resets the DLI vector register 512 as a back-up should the VBI routine be suspended during data transfer when downloading data to the printer. A deferred VBI routine is often suspended during data transfer so an immediate VBI routine has been employed instead. All this results in a much reduced screen flicker when printing out a league table. The VBI routine is defined as VBI$ on line 1870.

USING A DIFFERENT PRINTER

The program was written for use with an Epson compatible printer but it may be modified to use any printer. The printer control codes may need to be altered accordingly and are to be found on lines 820 and 840.

For reference when changing the control codes, the league table should print out using normal size characters with the exception of the heading "LEAGUE TABLE" which should print using double width, single height characters in the centre of an A4 sheet at the top of the page. The only other deviation is that the league tables category heading between the words 'POS' and 'DIFF' (inclusive) should be underlined.

A STRANGE ENDPROC

An ENDPROC with nowhere to go! At the end line 920 of the program listing, there is a strangely placed ENDPROC which can never be acted upon because it follows a GOTO (GO# NODATA). This ENDPROC is simply being used to reset Turbo BASICs indentation since there is no direct exit via an ENDPROC within the RESULTS procedure which can only exit via a POP command back to the menu from within the ENTER procedure. The ENDPROC is not an essential part of the listing but without it, Turbo's indentation is 2 places indented for the rest of the listing which gives the incorrect impression that something is wrong within the listing which Turbos indentation is supposed to identify.

PROGRAM BREAKDOWN

PROCEDURES

BEEP	Generates the beep
CLEAR	Clears the menu marker
CONTINUE	Used with the CREATE procedure requiring 'Y' or 'N' to be pressed to continue
CREATE	Creates new blank league tables
DISPLAY	Displays the league tables on screen and downloads the data to a printer
ENTER	Controls the entering of team names
FILES	Reads the disk's directory and displays all data files found on screen
INIT	Initialising routine
KEY	Gets a key press from the keyboard
LOAD	Loads a data file from disk
MARK	Marks the menu options when selected in inverse
RESULTS	For entering the individual results of the matches played
SAVE	Saves a data file to disk
SORT	Sorts the data in memory into order
SWAP	Used within the SORT procedure to swap over adjacent team names and the respective data if the lower team has a greater number of points or when teams are on the same number of points, the better goal difference

LINE LABELS

MENU	Start of the menu
NODATA	Start of the NO DATA routine
SELECT	Start of the 'select from menu' Routine

STRINGS

B$	Defined with 15 spaces following an inverse space to erase team names when necessary and providing a new cursor
F$	Defined as 'LEAGUE.DT*' (meaning league data) where '*' is the numeral selected when saving files to disk
I$	Used in the ENTER procedure to store the data as a team's name is being entered
LD$	Line (down), 40 Cntrl-M characters
LU$	Line (up), 40 Cntrl-N characters
N$	Dimensioned to 16 characters and normally used to identify team names from T$
P$	Used when printing out league tables
T$	Simulated string array for storing all the team names in a league
U$	General purpose string
X$	The screen RAM used for the extra screen lines
Z$	Defined with zero characters (the heart) used for erasing text from the screen

THE ARRAY

A(V,H): Where 'V' (vertical) represents the number of teams in the league minus one, that is on a scale of 0 to 23 and where 'H' (horizontal) represents the following league table data....

H=0	Games played
H=1	Points accumulated
H=2	Goals scored
H=3	Goals scored against
H=4	Home wins
H=5	Home draws
H=6	Home defeats
H=7	Away wins
H=8	Away draws
H=9	Away defeats

VARIABLES

AW	Identifies the away team
AWAY	The away team's score
HM	Identifies the home team
HOME	The home team's score
H,I,J,K,L	General purpose variables
NUM	Stores the number of teams in the current league
OPT	The menu option
SCR	The address of the screen RAM DPEEK(88)
Z	The address of Z$ used when erasing text from the screen

Many of the program's constants have been converted into variables to conserve memory. The values of these constant variables never change and are easily seen since they are preceded with the letter 'N' thus N4=4, N8=8, N10=10, N100=100, etc.

AND FINALLY ...

Please don't get the impression that I am interested in football after writing this program! My only interest in football is that I am a Womble, born and brought up in Wimbledon, a stones throw from Plough Lane, the original home of Wimbledon FC and my wife was born and brought up in Brentford, her father an ex-Brentford player.

THE LISTING

The full listing can be found on this issue's disk. If you prefer to type in the listing a TYPO coded printed listing is available on request, see inside back cover for details.



PROGRAMMING

AUTOMATIC PROGRAMMING

H S Wood demonstrates the unique Atari Return Key Mode

The ATARI 8 bit is capable of writing it's own lines for a program and this technique can be very useful.

I have included a simple example program called RTRNDEM.BAS which has full REM's to indicate how the lines perform the 'magic'.

METHOD

Basically one clears the screen, makes a program line a few lines down the screen and prints the word 'CONT' on the following line. The cursor is then moved above the line, and value 13 is poked into location 842. The 'STOP' command causes the program to stop and the cursor then moves down the screen to the printed line which is entered into memory. The cursor continues to the 'CONT' line and the computer re-starts. Next the value 12 has to be put into location 842 and the program is again running normally. The END command ends the run.

The value 13 in 842 means 'Read from the screen instead of the keyboard' so that anything on the screen is input to the computer. This is mentioned in 'Mapping the Atari' and in 'Sams Programmers Reference Guide' both of which are probably unobtainable but may be in lending Libraries for reference.

PROGRAMS

If the program is made to 'LOOP' many lines can be produced and a good example of this is a routine called MCDATA.BAS which produces 'DATA' lines from a machine code program in memory. The example I have provided for making 'DATA' lines is a modified version of a 5 line program published in ATARI USER dated November 1986 and written by Jeff Davis.

My modifications are to make it easier to type the program (there is a lot of program for 5 lines) and also to allow memory addresses to be entered in either 'HEX' or 'DECIMAL'. HEX is usually easier to use when dealing with machine code.

To use MCDATA.BAS type 'RUN' and follow the prompts. The DATA lines made by the above program should be 'LIST'ed to disk so that the program itself is not included. Also MCDATA.BAS should be protected so that it does not get corrupted.

DRAWING GRAPHS

A different program might draw graphs and thus use a FORMULA. This formula can be on a single line and will be used every time the program is run. However if the formula is to be changed to draw a different graph the RETURN KEY MODE can be used to write the same line with a different formula.

I have a GRAPH drawing program which I call FUNCTN.BAS which is included on this Issue's disk. Check the program out to see how it works. When asked for 'X' values type -10 and 10. The formula already in the program is 'Y(I)=SIN(X).

Similarly a program might 'READ' from a file and if the FILENAME has to be typed each time the program is 'RUN' it becomes tedious. Instead a message can be printed to ask 'Do you want to change the Filename'. Typing 'N' will run again with the same filename while 'Y' will ask for the new filename and the program will write a line with the new filename. This filename will continue to be used until 'Y' is typed again.

OTHER USES FOR THIS TECHNIQUE

Other uses for this technique include deleting lines of a program after it has 'LOAD'ed. A long program which 'LOAD's a lot of machine code from 'DATA' lines can have the DATA lines deleted after they have served their purpose thus taking up less room in memory. There are many more possibilities and Atari users will be able to think of plenty. ●

The MCDATA.BAS program can be found on the Issue disk.

```
EI 1 REM ******************************
ES 2 REM *    RETURN KEY MODE DEMO     *
WJ 3 REM *     FOR NEW ATRI USER       *
DV 4 REM *        By H S WOOD          *
RW 5 REM *        APRIL 1998           *
EN 6 REM ******************************
NM 7 REM
HC 10 REM *** SET LINE No.=1000 ***
MQ 15 LINE=1000
EE 45 REM *** CLEAR SCREEN ***
ZG 50 ? "[ESC,CLEAR]"
NE 65 REM **  PRINT LINE No. etc IN  **
EB 70 REM ** CORRECT PLACE ON SCREEN **
IP 75 POSITION 2,3:? LINE;" DATA ";
EL 90 REM *** PRINT ITEMS SEPARATED ***
CF 95 REM ***        BY COMMA's      ***
NC 100 FOR I=0 TO 5
VL 110 ? I;
DC 115 REM ** COMMA TO SEPARATE ITEMS **
OW 120 ? ",";
FX 130 NEXT I
YS 135 REM ** NO COMMA AFTER LAST ITEM **
DN 140 ? I
AH 200 REM ***    NOW THE 'MAGIC'    ***
TL 205 REM *** PRINT 'CONT' ON LINE ***
UY 210 REM ***   AFTER  DATA LINE    ***
SC 220 ? "CONT"
JC 225 REM *** MOVE CURSOR TO TOP ***
JK 230 REM ***     OF SCREEN      ***
HT 235 POSITION 2,0
EH 240 REM *** SET 'RETURN KEY' MODE ***
NN 245 POKE 842,13
OR 250 REM *** STOP THE PROGRAM ***
UH 255 STOP
HL 260 REM *** RESTORE 'NORMAL MODE' ***
NC 265 POKE 842,12
DE 270 REM *** CLEAR THE SCREEN ***
BW 275 ? CHR$(125)
HL 280 REM *** LIST NEW LINE AND END ***
SV 285 LIST 1000:END
```

Underline = INVERSE CHARACTERS · [ ] = CONTROL + CHARACTER · < > = INVERSE CONTROL + CHARACTER

A SHORT HISTORY OF COMPUTERS

A FUTURA Update by Austin Hillman

The electronic stored program computer is fifty years old this year. As you may know, the prototype of all modern computers 'Baby' has been reconstructed in time for its anniversary on 21st June 1998. But what came before it, and what came after?

EARLY CALCULATORS

The first calculating machine is said to have been built by Wilhelm Schickard around 1623, but it was later destroyed in a fire. Luckily, he described its workings in correspondence with Johannes Kepler, thus enabling a reconstruction to be made of his Calculating Clock. This consisted of a version of Napier's Bones, for multiplication, sitting on top of the mechanical device which carried out addition and subtraction.

Blaise Pascal created 'The Pascaline' mechanical calculator in 1642, to aid his father who was a tax collector. It was a shoe box sized device that could add, subtract, divide and multiply, but it was mechanically unreliable so only a dozen or so were made, some of which still exist.

In the 1670's Gottfried von Leibnitz created a major improvement to the mechanism of calculators with the invention of the Leibnitz Wheel which speeded multiplication. Sadly, his own calculator, the Stepped Reckoner, was apparently never perfected, as the only surviving version is inoperative.

COMMERCIAL PRODUCTION

The first commercially produced calculator using the Leibnitz Wheel, The Arithmometer, was created by Charles Xavier Thomas De Colmar for use in his insurance company. It appeared in 1820, and around 1500 examples were sold over the next thirty years or so thank to the new demands of the industrial revolution.

The most famous name in mechanical calculator design is of course Charles Babbage. He designed his Difference Engine in 1821, in order to compile logarithm tables. Unfortunately its complexity defeated the engineering of the day and it was abandoned in 1842.

However, in Sweden, Edvard Scheutz managed to build a working Difference Engine in 1843, based on the work of his father, Pehr. He built another in 1854 but found little demand for this remarkable device.

Undeterred, Babbage next designed the Analytical Engine in association with Augusta Ada, Lady Lovelace. This was to be the worlds first programmable calculator. The machine had an input where numbers were entered on punch cards, an idea borrowed from the Jacquard loom. The store held the numbers as required. The mill performed the arithmetic. The output printed the answer. The control unit was programmable for the type of calculation by punch card. Many different designs were drawn up over the years but the machine itself was never built.

In 1885, Dorr E. Felt developed a calculator that was operated only by the action of pressing the keys. It was called the Comptometer, and was advertised as "the machine-gun of the office" in order to emphasise the speed of operation.

In 1892, William S. Burroughs patented his Adding and Listing Machine which also used a keyboard. It was not key driven, but it was the first to produce a print out of the results. It became a best seller.

The American census of 1890 was processed by a punch card tabulator created by Herman Hollerith, who joined with others to create the company which would become International Business Machines (IBM) in 1911.

ENTER ELECTRONICS

In 1930, Dr. Vannevar Bush of the Massachusetts Institute of Technology produced a Differential Analyser, based on a paper written by Lord Kelvin in 1876. Copies of this device were built at several universities, including a Meccano version built at Manchester in 1935. It was an electro-mechanical analog computer designed to solve differential equations, but it was still basically just a calculator, albeit a powerful one. However things were about to change.

In 1937, the brilliant mathematician Dr. Alan Turing published his paper 'On Computable Numbers with an Application to the Entscheidungsproblem'. This paper lay out the theoretical concept of a machine that we now call a computer. All that was needed was someone to build a practical device.

Research mathematician George Stibitz created his Model K (for Kitchen) computer in 1937. This was a device using relays that could add binary numbers. Developing the idea at Bell Labs, his Complex Number Calculator was completed in 1939. Bell was not

Wilhelm Schickard's first calculating machine — 1623
The Pascaline — 1642
Leibnitz Wheel — 1670

very interested in developing it, but the U.S. Army was, they took delivery of five relay computers for ballistic computations. This range reached its peak with the BTL6 of 1949, although research continued into the 1950's.

Meanwhile in Germany, engineering student Konrad Zuse began work on a binary computer in 1936, to aid his work at Henschel Aircraft. The Z1 was a mechanical demonstration model using switches and bulbs. The Z2, completed in 1939, used relays. Military funding for a code breaking model was refused in 1940. Continuing to work almost unaided, Zuse and colleague Helmut Schreyer, built the Z3 in 1941. This used 2600 relays and was controlled by instructions on punch tape. The partly electronic Z4 was completed in 1945. His company, Zuse AG, founded in 1949 would form the basis of the German computer industry, becoming part of Seimens in 1967.

Back in America, Harvard engineer Howard Aiken had designed a relay computer, based on the ideas of Babbage, in 1937. He persuaded IBM to build the Harvard Mark I Automatic Sequence Controlled Calculator in 1939. It was finally completed in 1944. The Mark II arrived in 1947. The Mark III in 1950 and Mark IV in 1952. These relay computers were large, expensive, and only ten times as fast as a mechanical calculator, but they were reliable and could work 24 hours a day.

A 30 TON COMPUTER!

During the war America was desperate for accurate ballistic tables. Physics Professor John Mauchly proposed an electronic computer in 1942 but it was not until April 1943 the Army sanctioned the building of the Electronic Numerical Integrator Analyser and Computer - ENIAC. It finally entered service in February 1946. It was a huge device containing 18000 valves and weighing 30 tons. It could handle numbers of up to 20 digits and hold 10 of these numbers in its store. It was also very fast, up to 5000 calculations per second.

It had been thought that ENIAC was the first electronic programmable computer. However the UK government revealed in October 1975 that an electronic computer, code name Colossus, was operational at Bletchley Park in December 1943, and that ten units were operational at the end of the war. They replaced relay based machines that had been used from 1941. These 1500 valve code breaking machines were built by Professor M.H.A. Newman and T.H. Flowers, based on the ideas of Alan Turing. Development was continued on the Colossus design at the Radar and Telecommunications Research Establishment until 1963.

Work began on the successor to ENIAC, the Electronic Discrete Variable Automatic Calculator - EDVAC, in 1946. It was eventually completed in 1951, as in the meantime John Mauchly and J.P.Eckert had left the project to form their own computer company. After creating BINAC for Northrop Aircraft in 1948, they produced a commercial computer, UNIVAC 1, in 1951, the first to use magnetic tape for storage of data. Remington Rand bought the company in 1952 and sold 48 units, the last of which was turned off in 1970.

Mathematician John von Neumann had worked with Alan Turing in the late 30's. He published an influential paper on the design of a stored program computer in 1946. After working on EDVAC he created the IAS computer at the Institute of Advanced Studies at Princeton in 1952. This machine was to heavily influence future computer design.

At the National Physics Laboratory, Turing designed the very powerful Automatic Computing Engine - ACE, before joining the Manchester University team in 1948. ACE was eventually built in 1957 after the less ambitious Pilot ACE had been constructed. This was to form the basis for the English Electric range of commercial computers.

THE FIRST REAL COMPUTER

The first stored program computer was therefore the Manchester University Mk 1 'Baby', created by Professor Frederick Williams and Doctor Tom Kilburn. This 600 valve unit, capable of processing 800 instructions per second, used a memory device known as a Williams tube. This was basically a cathode ray tube adapted to store 128 bytes of information. The Manchester team later collaborated with others to produce a commercial version, the Ferranti Mk 1 in 1951, which lead to the Pegasus of 1959, followed by the Mercury and then the Atlas, the fastest computer available in 1963.

Cambridge University produced EDSAC - Electronic Delayed Storage Automatic Calculator, a scaled down version of EDVAC, in 1949. The Cambridge team also assisted with the design of the Lyons Electronic Office, a massive 6000 valve unit, whose 'memory' was a collection of mercury tubes weighing half a ton. This was developed into the LEO 1 commercial computer of 1954. Leo Computers Limited built about 100 units and became part of English Electric in 1964.

Birkbeck College produced experimental models based on the IAS design, which were the inspiration of the commercial units made by the British Tabulating Machine Co., now known as International Computers Limited.

The mighty IBM despite building the Harvard computers was not really interested in this area until it was approached by the US government. It built the IBM 701, based on the IAS design, in 1953. It was surprised to get 18 orders for this model. Now convinced there was a demand for computers, the IBM 702 and 704 were launched in 1955 for commercial use.

FIRST USE OF TRANSISTORS

Transistors, invented in 1947, were first used in the TX-O (Transistor eXperimental cOmputer) built at MIT in 1956. They finally supplanted the mighty valve in the IBM 7000 series of 1959.

Integrated circuits, created in 1958, first appeared in the IBM System/360 range of 1964. The BASIC programming language was invented by John Kemeny and Thomas Kurtz that year.

The Intel Corporation, founded in 1968, pioneered the development of memory chips. And it was here that Marcian E.Hoff produced the first microprocessor, the 4004 (a 4-bit processor intended for use in a pocket calculator), in 1971.

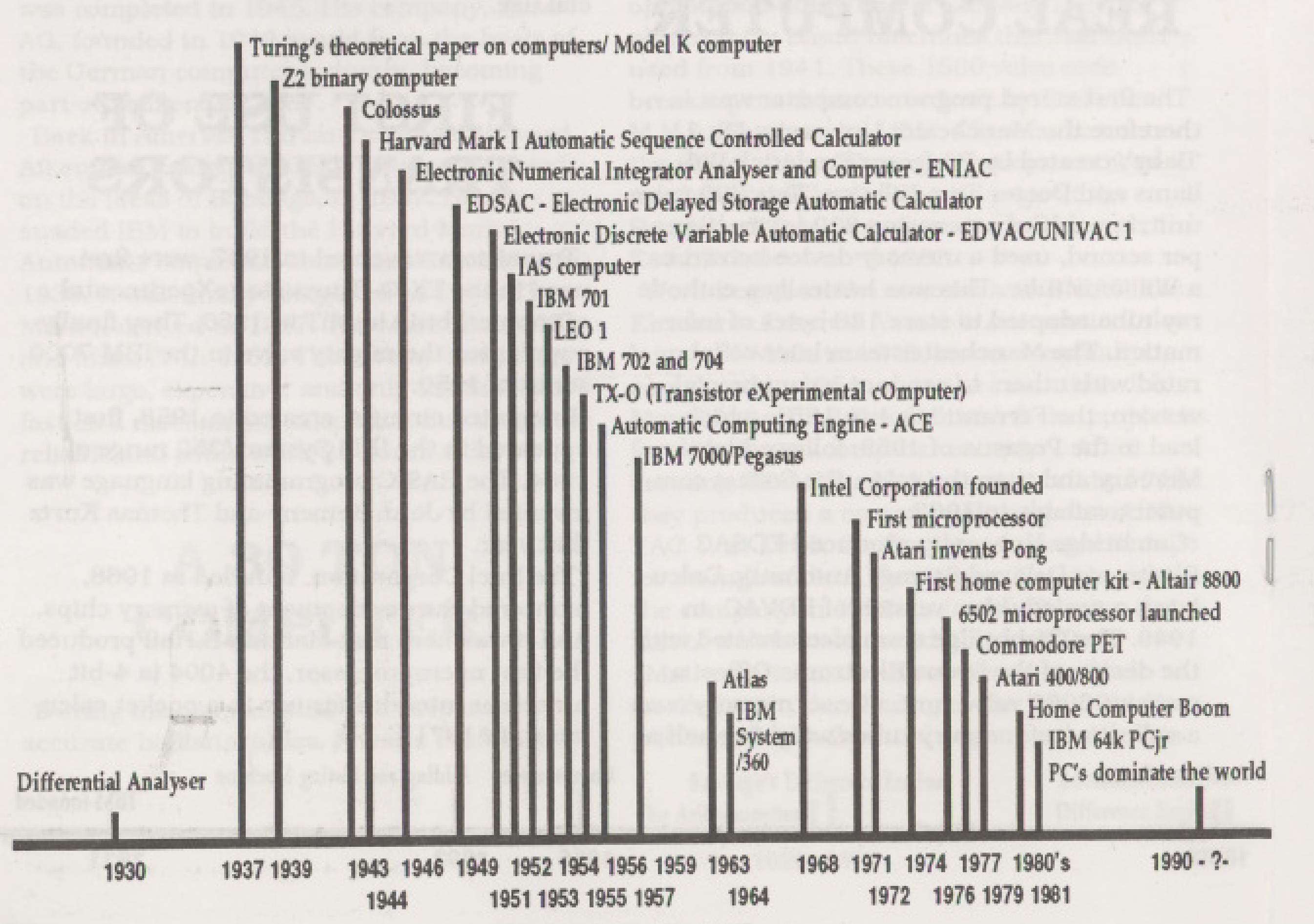
The Arithmometer 1820 | Babbage's Difference Engine 1821 | Edvard Scheutz's Difference Engine 1843 | Analytical Engine 1850's | Comptometer 1885 | Adding and Listing Machine 1892 | IBM founded 1911

THE HOME COMPUTER

The first home computer was the the Altair 8800, launched as a kit in December 1974, by Micro Instrumentation and Telemetry Systems. It used the Intel 8080 eight-bit microprocessor (1973) and had a minuscule 256 bytes of ram to work with. Basically a large box with some switches and pretty lights on the front it could do very little. It cost $397.

In 1976 the 6502 microprocessor was launched by MOS Technology (based on the Motorola 6800 of 1974) and used in their KIM 1 (Keyboard Input Monitor 1) kit computer. The success of this crude device supposedly encouraged Commodore to launch the PET (Personal Electronic Transactor) in June 1977.

The home computer boom was now just around the corner, the Atari 400 and 800 would soon arrive to do battle with a flood of micro's that came and went in the 80's. British proved to be best as the Sinclair range saw off allcomers, including the Japanese with their compatible MSX models. The IBM 64k PCjr of 1981 initially seemed to be another failure but after a slow start the PC and its many clones have emerged as the winners of the hardware battle - at least for now.



TEST CARD

John Foskett helps you set up your TV for the perfect picture

The test card utility program was written as an aid for television and monitor alignment to ensure that they give the best possible display. This program, a test card or any pattern generating equipment is used to provide a standard image on screen to give a visual indication when setting up the internal adjustments of a television set or a monitor.

CAUTION!

At this point a cautionary note is necessary, television and monitor alignment is a skilled job which should NEVER be attempted without prior knowledge. Such equipment contains very high voltages which can be dangerous so ALWAYS be careful. Remember at all times that electricity cannot be seen, heard, smelt or tasted, it can only be felt and once felt it might be too late, so treat electricity with the respect it deserves!

POOR QUALITY PICTURES

It is really amazing how so many people put up with poor quality pictures when watching television, but much of the time they are unaware of the problem simply because the quality of a picture deteriorates very slowly. It is extremely difficult to notice misalignment when watching normal television pictures, but the problems show up clearly when using a test card or a pattern generator.

USING THE PROGRAM

The Test Card program provides a grid of horizontal and vertical lines which are evenly spaced with a large circle in the centre. This is used to ensure a good linear display such that the lines are evenly spaced over the entire screen both horizontally and vertically and also that the circle is circular and not elliptical in any way. The lines should appear straight without any curvature or kinks. Within some of the boxes formed by the grid and spread evenly over the screen are small

```
PM 10 REM ****************************
DK 20 REM *   THE TURBO TEST CARD    *
VS 30 REM *  FOR MONITOR ALIGNMENT   *
HR 40 REM * WRITTEN BY JOHN FOSKETT *
DO 45 REM * ------------------------ *
KL 50 REM *  NEW ATARI USER - 1998  *
PR 60 REM ****************************
BN 65 REM
GF 70 N24=24:GRAPHICS N24:POKE 16,64:POKE
    53774,64:POKE 710,%0:N6=%3+%3:COL=%0:
   EXEC COLOUR:N256=256
MD 80 RESTORE 380:FOR I=0 TO 33:READ J:PO
   KE 1536+I,J:NEXT I
BJ 90 RESTORE 90:FOR I=%0 TO N6:READ J:PO
   KE DPEEK(560)+J,143:NEXT I:DATA 10,40,
   64,88,114,138,162
PL 100 POKE 54286,%0:DPOKE 512,1536:DPOKE
    548,1559:POKE 54286,192:DIM A$(N256),
   B$(N256)
KQ 110 A$=CHR$(255):A$(N256)=CHR$(255):A$
   (%2)=A$:B$="[,]":B$(N256)="[,]":B$(%
   2)=B$:B$(46)="?":B$(210)="?":B$(47)=B$
   (46)
TP 120 FOR I=%0 TO N6:J=I*N24+44:B$(J,J+%
   1)="[,][,]":NEXT I
QF 130 REM  Set-Up PMGs
JN 140 I=PEEK(106)-40:POKE 54279,I:PMB=I*
   N256:POKE 53277,%2:POKE 623,8:POKE 704
   ,10:POKE 705,10
ZU 150 FOR I=%0 TO %3:POKE 53248+I,%0:POK
   E 53256+I,%1+(I(%2)*%2:NEXT I
JT 160 FOR I=%0 TO %1:MOVE ADR(A$),PMB+10
   24+I*N256,N256:NEXT I
WV 170 FOR I=%0 TO %1:MOVE ADR(B$),PMB+15
   36+I*N256,N256:NEXT I
ME 180 REM  Print Graphics
BX 190 COLOR %1:FOR I=11 TO 191 STEP N24:
   FOR J=%0 TO %1:PLOT %0,I+J:DRAWTO 319,
   I+J:NEXT J:NEXT I
OO 200 FOR I=%3 TO 319 STEP N24:FOR J=%0
   TO %1:PLOT I+J,%0:DRAWTO I+J,191:NEXT
   J:NEXT I
CI 210 FOR I=15 TO 319 STEP 48:FOR J=23 T
   O 190 STEP 48:PLOT I,J:PLOT I+%1,J:PLO
   T I,J+%1:PLOT I+%1,J+%1:NEXT J:NEXT I
YS 220 FOR I=159 TO 160:FOR J=95 TO 96:CI
   RCLE I,J,95:NEXT J:NEXT I
LW 230 REM  Print Text
WN 240 TEXT 40,%0,"THE TURBO":TEXT 208,%0
   ,"TEST CARD":TEXT 16,184,"JOHN FOSKETT
   ":TEXT 215,184,"MARCH 1997"
RF 250 REM  Turn On PMGs
GX 260 POKE 559,58:RESTORE 260:FOR I=%0 T
   O %3:READ J:POKE 53248+I,J:NEXT I:DATA
    15,209,106,130
YS 270 REM  Options
EA 280 DO :I=PEEK(53279):IF I=N6:COL=COL+
   16:IF COL)240:COL=%0:ENDIF :EXEC COLOU
   R:PAUSE 15:ENDIF
NC 290 IF I=%3 THEN COL=%0:EXEC COLOUR
JU 300 IF I=5 THEN J=22:FOR I=%0 TO N6:J=
   J+32:POKE 590+I,J:POKE 606-I,J:NEXT I
SM 310 IF PEEK(732)=17 THEN POKE 732,%0:B
   ACK=148-BACK:POKE 712,BACK:POKE 710,BA
   CK
RA 320 LOOP
JK 330 --
VW 340 REM  Load Colours
RJ 350 PROC COLOUR:J=COL:FOR I=%0 TO N6:J
   =J+%2:POKE 590+I,J:POKE 606-I,J:NEXT I
   :ENDPROC
JQ 360 --
UE 370 REM  DLI Data
EN 380 DATA 72,138,72,166,203,230,203,189
   ,78,2,141,20,208,189,88,2,141,21,208,1
   04,170,104,64
XA 390 REM  VBI Data
LK 400 DATA 72,169,0,133,77,133,203,104,7
   6,138,194
```

Underline = INVERSE CHARACTERS - [ ] = CONTROL + CHARACTER - < > = INVERSE CONTROL + CHARACTER

PMG PROGRAMMING TIPS

by John Foskett

When using PMGs, it is normal practice to lower RAMTOP to provide a protected area of RAM in which to store the shape data and conventionally for single line resolution, PMGs require 2 kBytes or 8 pages of RAM. In actual fact the PMG data only requires 5 pages of RAM for storage because the 3 pages of RAM immediately below the missile shape data is always unused and is therefore wasted if RAMTOP is lowered by 8 pages. It is possible however to protect only the required 5 pages of RAM leaving the 3 pages of unused RAM available for BASIC which is achieved using....

```
RAMTOP=PEEK(106):POKE
106,RAMTOP-5:PMBASE=RAMTOP-8
```

If you lower RAMTOP by the conventional 8 pages and you are not using the missiles, then you have 4 pages of protected RAM immediately below the player shape data free for use, ideal for storing a whole new redefined character set.

If you are only using one player in a program, then use PLAYER 3, the highest in RAM so that you only need to protect a single page of RAM for its shape data storage. This will allow you a whole 7 pages of extra free RAM. It then follows that if you use 2 players in a program, using PLAYERS 2 and 3 will save 6 pages of RAM, etc.

Alternatively, you don't have to lower RAMTOP to protect your PMG data (or any data for that matter), you could allow BASIC to encroach into the data area if you wish. The great advantage in doing this is that you will immediately notice when you have used up all of the available RAM since it will show up on screen as rubbish in the PMG stripes. But you must however define your PMG shapes early in a program before defining strings otherwise the PMGs will overwrite the strings instead. Not lowering RAMTOP in this way will avoid the "Out of Memory" error 2 from occurring which can cause a crash. Loops and subroutines need RAM in which to function and if insufficient RAM is available for this, then stack errors can occur when a program is run causing a crash and therefore the loss of your program. ●

TEST CARD continued

spots which are used to ensure accurate focusing, all the spots as well as the rest of the screen should appear clear and sharp. It is much easier to set the focus by using small spots rather than by using lines. Initially in the centre of the circle is a grey scale from black to white which is used to set up the brilliance and contrast adjustments and the colour balance to give the full range of greys without any colour tinge. Pressing START will slowly cycle through the colours giving colour scales to ensure a good colour range and balance. Pressing OPTION returns the grey scale and pressing SELECT displays a range of colours at a mid-brilliance setting. Pressing HELP toggles the background colour from the initial black to the normal Atari blue which more easily shows up the flyback lines and also tuning inaccuracies. When blue, the screen should be clear without any background roughness or flyback visible. At the sides of the screen are two vertical stripes to aid the horizontal and vertical positioning of the display within the boundaries of the screen to ensure a perfectly centred picture. ●

HEY! HEY! it's

The TIPSTER

This issue our regular Tipster James Matthrick concentrates on some Public Domain adventures available from the Page 6 Library. There are hundreds, if not thousands, of PD games available of all kinds so how about some hints and tips for these for future issues? The future is here, it may not be orange but it could be PD!

LIVINGSTONE

Page 6 Adventure Set. #1 Disk 2A
Page 6 Library disk #30

The mapping system for the game is unnecessarily complicated, and detracts from the game, however there is a system to it - you will need to be patient and persevere with this game. Some hints to help you on your way:

In the bedroom, wear the boots and the knapsack, open the book, read the book in bed, and type **DRIFT**.
Should you find your path blocked by quicksand, merely **JUMP QUICKSAND**.
Catch the mouse in your knapsack, then free the mouse when confronted by the leopard.
When you have the diamond, if you **SAY SWAMI** you will find yourself back in your room.
Catch the viper, and free it near the dog. The viper is in the tree.
You may find yourself trading with the natives, however, do not take the spear into the native village - the natives will take it as being a threat.

Maybe someone could come up with a map or a complete solution - I really lost enthusiasm for this game because of the mapping system. A Tipster challenge maybe?

DUNGEON OF THE GODS

Page 6 Adventure Set #1 Disk 1B
Page 6 Library disk #23

The aim of the Game is to escape from the Dungeon of the Gods - here are some hints and tips to complete the game, in no particular order:

OPEN BOX in the first room whilst holding the bar

POTION 1 and **POTION 2** will make you either weaker or stronger - use carefully

INSERT the copper **COIN** when in the casino

WAND 0 will shoot lightning bolts, use it against the slime, but not the black dragon

To use the magic dust, **WAVE DUST**

Use the **LANCE** against the **BLACK DRAGON**, and the **FLAMING SWORD** against the **BLACK CUBE**

Collect the **PLATINUM BAR**, the **DIAMONDS**, and the **CHEST OF THE GODS** before you escape for maximum points

READ the **SCROLL** at the **RED DRAGON**

THROW the **BALL** at the **STATUE**

POUR the **ACID** at the entrance

WAVE WAND 2 at the **STEEL DOOR**

PUSH BUTTON in the **STRANGE RED ROOM** to get to 'the west end of a prison cell'



Dungeon of the Gods

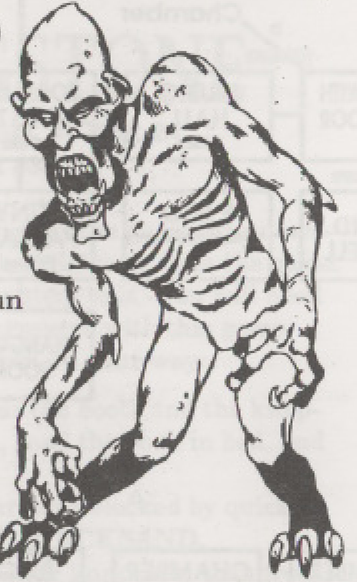
LIBRARY Scroll
DARK ROOM Slime Creature Ring
LAB Acid
RECTANGULAR ROOM Ball
KITCHEN
CASINO Slot Machine
D
To Dirty & Dusty Room
DRAGON ROOM Diamonds
DARK HALL Black Cube
WORKSHOP Wand 2 Table
SQUARE ROOM
To Tiny Hall
U
To Small Room
U
STAIRS
LARGE HALL Sword
STORAGE ROOM
STAIRS UP
THRONE ROOM Statue
ROOM WITH IVORY TABLE Wand 0
WIZARD'S WORKSHOP Magic Dust
TREASURE VAULT Snake/Platinum
To Small Chamber
D
SMALL ROOM Potion 2
ROOM WITH STEEL DOOR
MUSIC HALL Flute
ROOM WITH A BRASS TABLE Coin
WEAPON ROOM Shield/Lance
STORAGE ROOM Lamp
W END OF CELL
E END Bar/Box/Key
TINY PASSAGE Potion 1
TRIANGULAR ROOM
DIRTY & DUSTY ROOM
CHAMBER OF THE GODS Chest
CHAMBER OF DEATH
BRIGHT CHAMBER
VERY LONG CORRIDOR
END
HUGE ENTRANCE
To Stairs Up
D
SMALL CHAMBER
DARK CORRIDOR
U
To Room with Steel Door
TINY HALL
LONG DARK CORRIDOR
SMALL ROOM
D
To Stairs
STRANGE RED ROOM Button
To W End of Cell

More Tips ☞

QUEST

Page 6 Adventure Set #1
Page 6 Library disk #23

Solution: Cut across country, mind your own business, kill the hobgoblins, search the hobgoblins (find something!), go south, go south, reason with the dwarf, yes, 7,5,9, go south, go castle, go door, say something to the wizard, go door, attack, go cave, run to back of cave, read runes, say word, unlock door, lift stone door, go into room...



COLONY

The following tips for Colony from Bulldog/Mastertronic were discovered on my Hard Drive where they were obviously transferred to from an 8-bit disk but I can't credit the Tipper. My apologies and thanks to whoever you are. If you care to admit 'It was me' I will happily give you a mention in the next issue. Anyway here are the tips for Colony:

Want lots of money fast? Order lots of extra seeds from the supply ship, and as soon as it arrives take them straight to the mushroom/seed store, but don't drop seeds. Instead, deposit mushrooms, and the computer will think you are depositing mushrooms ten at a time, and credits your account accordingly. This doesn't always seem to work, but does most of the time. I think you have to deposit the seeds before you do anything else, once the supply arrives, but I'm not positive.

HELP ... HELP ... HELP

This request comes form the same unknown source as the Colony tips but if you have the answers, I am sure they will be of interest to everybody so write in. Help please with **PANTHER** (Mastertronic). Can't get past the plasma wave. Too bloody impossible, and seems to last forever. Can it be done, and if so, how?

Remember The Tipster needs YOU to stay alive. Write now to:

THE TIPSTER
NEW ATARI USER
P.O. BOX 54
STAFFORD
ST16 1DR

TURBO TUTORIALS

Gavan Moran brings you a few examples to illustrate the power of Turbo programming

PAC DEMO

This demo is primarily an example of how the MOVE command of Turbo Basic can be used to animate player missile graphics. A few other features of Turbo Basic are also used and provide a good example of the way in which Turbo Basic simplifies programming.

WHAT IT DOES

An animated 'Pacman' figure is chased across the screen, from right to left, by a ghost. The pacman eats a 'power pill' situated on the left hand side of the screen. At this, the message 'TURBO POWER' appears in large letters across the screen, the ghost changes colour to blue and the pacman chases the ghost back across the screen, eventually eating it.

HOW IT WORKS

Lines 190 to 410 contain the setup procedure. Player/missile graphics are used to represent the pacman, ghost and power pill. The playfield is a GRAPHICS 19 (3+16) screen.

The routine to draw the 'turbo power' message is contained in a separate procedure (lines 150 - 170) and is called by the setup procedure. The TEXT command is used to plot the large letters - these can be quite effective.

Note the use of hexadecimal numbers in line 280. This is a helpful feature of Turbo Basic as it allows direct use of hex numbers in programs - useful for machine language programmers.

A DO - LOOP loop is used to read the P/M data into memory. An EXIT command is used to escape from the loop when a -1 is encountered in the data. The data is read into the normally unused memory area between PMBASE and the missiles. A number of animation 'frames' may be stored here (up to 384 bytes in double line resolution or 768 bytes using single line).

Back to the main program which is contained within a DO - LOOP loop which continuously executes the main body of code.

Two FOR - NEXT loops control the movements of the players from right to left and left to right respectively. Both these routines call the MOVE-PLAYERS procedure and it is here that the real beauty of the MOVE command becomes apparent.

As I said earlier, frames of player data are stored in the unused part of player memory. The MOVE-PLAYERS procedure uses the MOVE command of Turbo Basic to transfer these into the appropriate player - thus we have a simple but effective player/missile animation system! If this sounds familiar then I recommend you read the article 'PLAYER MISSILE GRAPHICS' in issue 15 of PAGE 6 - the technique used there is virtually identical. Who said Turbo Basic had no player/missile commands?

So there we have it - a simple but effective Turbo Basic demonstration which would probably have taken at least three times as much source code in normal Basic, not to mention a machine language subroutine. Hopefully the code is easier to read than normal Basic too, but I had to compress it somewhat to make it fit into one page. Remember, MOVE can be used for other tasks too - such as moving character sets or display lists.

TEXT PLOTTER

This program hopefully illustrates the power of some of the more unfamiliar Turbo Basic commands. Also, I hope, it should be useful to those people who use ATARI ARTIST or MICROPAINTER software as it allows the user to add text to pictures which are saved in the standard 62 sector picture format.

USING THE PROGRAM

The program uses a mixture of both joystick and keyboard commands. Initially, when RUN, a blank screen will appear with four small orange dots in the upper left hand corner. This is the cursor - use the joystick to move it around the screen. It denotes the position of any text that will be printed. You may add text by pressing the fire button and then typing in your text, terminated by a <RETURN>. To save this screen press the greater than (>) key. The screen will then be saved as an uncompressed 62 sector file called D:PICTURE to which a picture may be added using an art package at a later date. Most likely, you will want to add text to an existing picture and this can be accomplished by using the less than (<) key to load a 62 sector file called (you guessed it) D:PICTURE from your disk. Note that this loaded file will overwrite anything already on the screen. Text may then be added as above and the screen re-saved. Use keys 0..4 to change the colour register being used to plot the text. The cursor will change colour accordingly to let you know which register is being used (they correspond to the registers on the bar at the bottom of the ATARI ARTIST menu). The escape (ESC) key may be used to clear the screen. Note that the entire character set is available for plotting (including control characters) but that the standard editing keys do NOT work - to delete a character, move the cursor back over it and type a space.

HOW IT WORKS

The main program is contained in the DO - LOOP loop. This repeatedly checks the joystick and the keyboard and calls appropriate procedures if action is needed. Note the use of

```
QK 1 REM ****************************
MX 2 REM *  TURBO BASIC - PAC DEMO  *
HJ 3 REM *      by Gavin Moran      *
IY 4 REM * ------------------------ *
VF 5 REM *  NEW ATARI USER - 1998   *
QP 6 REM ****************************
NM 7 REM
OW 10 EXEC SET_UP
IZ 20 DO
YJ 30 MOVE PM+55,PM+826,11:POKE HPOSP2,50:POKE COLP0+%1,72:SETCOLOR 0,0,0:SETCOLOR 1,0,0:PAC=PM:OFFS=12
XE 40 FOR X=200 TO 50 STEP -5:EXEC MOVE_PLAYERS:NEXT X
RU 50 POKE HPOSP2,0:POKE COLP0+%1,132:SETCOLOR 0,4,8:SETCOLOR 1,0,15:PAC=PM+11:PAUSE 25
ME 60 FOR X=50 TO 200 STEP 5:EXEC MOVE_PLAYERS:OFFS=OFFS-0.35:NEXT X:PAUSE 100
YE 70 LOOP
ZD 80 --
PF 90 PROC MOVE_PLAYERS
CW 100 MOVE PAC,PM+570,11:MOVE PM+33,PM+698,11:PAUSE 7
NE 110 MOVE PM+22,PM+570,11:MOVE PM+44,PM+698,11:POKE HPOSP0,X:POKE HPOSP1,X+INT(OFFS)
YM 120 PAUSE 7
VP 130 ENDPROC
JK 140 --
ZM 150 PROC TURBO_LOGO
EN 160 COLOR 1:TEXT 0,3,"TURBO":COLOR 2:TEXT 0,13,"POWER"
VX 170 ENDPROC
JS 180 --
FI 190 PROC SET_UP
NB 200 RAMTOP=PEEK(106)*256
CX 210 RAMTOP=RAMTOP-1024
ES 220 POKE 106,RAMTOP/256
WI 230 GRAPHICS 19:SETCOLOR 0,0,0:SETCOLOR 1,0,0
CJ 240 FOR X=RAMTOP+512 TO RAMTOP+896:POKE X,%0:NEXT X
WC 250 EXEC TURBO_LOGO
MQ 260 PMBASE=$D407:GRACTL=$D01D:SDMCTL=$022F:GPRIOR=623:COLP0=$02C0:HPOSP0=$D000:HPOSP1=$D001:HPOSP2=$D002
AJ 270 RESTORE #PMDATA
VD 280 OFFSET=%0:PM=RAMTOP
WC 290 DO
IR 300 READ PMBYTE
SW 310 IF PMBYTE=-1 THEN EXIT
HW 320 POKE PM+OFFSET,PMBYTE
DA 330 OFFSET=OFFSET+%1
RE 340 LOOP
SI 350 POKE PMBASE,(RAMTOP/256)
NS 360 POKE GRACTL,3
EG 370 POKE SDMCTL,PEEK(SDMCTL)+12
BL 380 POKE COLP0,248:POKE COLP0+%1,72:POKE COLP0+%2,15
WD 390 ENDPROC
JF 400 --
DQ 410 # PMDATA:DATA 60,126,247,127,63,15,63,127,255,126,60,60,126,239,254,252,240,252,254,255,126,60
ZC 420 DATA 60,126,247,255,255,255,255,255,255,126,60,60,126,255,153,221,255,255,255,255,171,0
UB 430 DATA 60,126,255,153,187,255,255,255,255,85,0,0,0,0,60,126,255,255,126,60,0,0,-1
```

Underline = INVERSE CHARACTERS - [ ] = CONTROL + CHARACTER - < > = INVERSE CONTROL + CHARACTER

```
QK 1 REM ******************************
EW 2 REM *TURBO BASIC - TEXT PLOTTER *
HJ 3 REM *      by Gavin Moran        *
IY 4 REM * -------------------------- *
VF 5 REM *   NEW ATARI USER - 1998    *
QP 6 REM ******************************
NM 7 REM
OW 10 EXEC SET_UP
IZ 20 DO
IX 21 EXEC CHECK_STICK
GP 25 IF PEEK(764)<>255:GET KEY
AE 40 IF KEY=60
ZO 50 EXEC LOAD_FILE
UL 60 ELSE
AT 70 IF KEY=61
GB 80 EXEC SAVE_FILE
UO 90 ELSE
PW 100 IF (KEY>47 AND KEY<52) THEN CURSOR
   =KEY-48
IK 110 ENDIF
IM 120 ENDIF
XS 130 POKE 764,255:ENDIF
RC 140 LOOP
JM 150 --
VE 160 PROC PLOT_CURSOR
NK 170 LOCATE XPOS,YPOS,PIXEL:LOCATE XPOS
   +7,YPOS,PIXEL1:LOCATE XPOS,YPOS+7,PIXE
   L2:LOCATE XPOS+7,YPOS+7,PIXEL3
HC 180 COLOR CURSOR:PLOT XPOS,YPOS:PLOT X
   POS+7,YPOS:PLOT XPOS,YPOS+7:PLOT XPOS+
   7,YPOS+7
MR 190 LX=XPOS:LY=YPOS
VK 200 ENDPROC
JF 210 --
WC 220 PROC LOAD_FILE
EV 230 DPOKE 88,SCREEN:POKE 87,15
DM 240 OPEN #1,4,0,"D:PICTURE"
FY 250 BGET #1,SCREEN,7680
LK 260 CLOSE #1
RW 270 EXEC PLOT_CURSOR
WA 280 ENDPROC
JV 290 --
CY 300 PROC SAVE_FILE
FH 310 OPEN #1,8,0,"D:PICTURE"
MD 320 BPUT #1,SCREEN,7680
LF 330 CLOSE #1
VT 340 ENDPROC
JO 350 --
LV 360 PROC CHECK_STICK
MI 370 S=STICK(0) EXOR 15
QA 380 IF S<>0
YI 390 XPOS=XPOS+(S&8>0)-(S&4>0)+(XPOS<%1
   )-(XPOS>151):YPOS=YPOS-(S&1>0)+(S&2>0)
   +(YPOS<%1)-(YPOS>181)
XR 400 COLOR PIXEL:PLOT LX,LY:COLOR PIXEL
   1:PLOT LX+7,LY:COLOR PIXEL2:PLOT LX,LY
   +7:COLOR PIXEL3:PLOT LX+7,LY+7
RM 410 EXEC PLOT_CURSOR
IP 420 ENDIF
DT 430 IF STRIG(0)=%0 THEN EXEC PLOT_CHAR
   ACTER
VU 440 ENDPROC
JP 450 --
WM 460 PROC PLOT_CHARACTER
MZ 470 GET CHAR
FV 480 WHILE CHAR<>155
PB 485 TEXT XPOS,YPOS,CHR$(CHAR)
LN 490 XPOS=XPOS+8:IF XPOS>152 THEN XPOS=
   152:EXIT
RL 500 EXEC PLOT_CURSOR
MO 510 GET CHAR
MY 520 WEND
VT 530 ENDPROC
JO 540 --
FE 550 PROC SET_UP
VJ 560 GRAPHICS 31:SCREEN=DPEEK(88)
UY 570 XPOS=%0:YPOS=%0:CURSOR=%1:EXEC PLO
   T_CURSOR
WD 580 ENDPROC
```

Underline = INVERSE CHARACTERS · [ ] = CONTROL + CHARACTER · < > = INVERSE CONTROL + CHARACTER

GET without needing an OPEN and the way CLS #6 is used to clear the screen. Procedures LOAD-FILE and SAVE-FILE use the BGET and BPUT commands to read/write the screen data from/to disk - much handier than machine code CIO calls and much quicker than the usual PUT/GET method. Procedure CHECK-STICK uses the binary manipulation facilities of Turbo Basic in conjunction with a little boolean algebra. Note the use of the EXOR (EXclusive OR) and '&' (binary AND) commands to read the stick directly. The result is a compact, quick method of joystick reading. NB. Touch tablet users may wish to change the code to read STICK(1) in order to use the second joystick port to avoid having to swap the tablet and the joystick. PLOT-CHARACTER uses the TEXT command to plot the text on to the graphics screen - another useful Turbo Basic command.

Well, that's about it. Because this program was written within the constraints of having to fit in a page, there are quite a few features missing. How about a P/M cursor (could be controlled using MOVE command), memory save (again using MOVE command), RAM-DISK support on 130XE or 256K 800XLs, multiple fonts and different text directions-?Plus the excellent PICLOADA code printed in issue 20 could be added in order to enable direct loading and saving of ATARI ARTIST screens. And of course, the program can then be compiled for even more speed!

MINI CHARACTER SET EDITOR

This program is a little too long to include in the magazine so you will find it on the Issue disk. I have not written a full set of documentation, however there follows some short instructions on how to use the program.

When the program is RUN, a status line at the top left hand corner of the screen informs you of the ASCII no, internal no and actual character being edited. Below these is a character grid showing the character which you are editing. A lit square on the grid represents a lit pixel in the character and a dot represents an unlit pixel. To the right of this is a large representation of how the character looked before you started editing it. Moving the joystick will cause a star shaped cursor to move around the grid. Pressing the fire button will toggle the pixel under the cursor either on or off. A list of commands is shown below and these are selected by typing the key which is highlighted in inverse text. Commands are: Load - loads a character set off disk. This set MUST be called CHSET.FNT. If this is not present the program will crash. Save - Saves the set currently in memory to the disk in a file called CHSET.FNT. Getchar (ASCII) - allows you to type the ASCII number of the character which you wish to edit, e.g. 65 will get 'A' to be edited on the grid. Getchar (type key) - allows you to get a character for editing by typing it's equivalent key on the keyboard, e.g. Typing 'H' in this option will bring the character H up for editing. Putchar - Puts the character being edited back into the set. This must be done in order to permanently change the character as the character is edited in a separate buffer. Default - Returns the character set to it's default (internal) state.

This program hopefully demonstrates how Turbo Basic simplifies programming complicated tasks and how it's increased power allows shorter code to be written. ●



DISK DIRECTORY MOVER VERSION II

by John Foskett

In response to a letter from James Austin in Mailbag of Issue 80 which made a few suggestions for updating my Disk Directory Mover published in issue 79, I present Disk Directory Mover Version II. In his letter, James suggested that there should be a means of going direct to the menu bypassing the formatting stage so that all the files do not have to be copied in one session. James also suggested that there should be a means of listing the directories of both the standard disk and the modified disk from the menu.

For a full explanation of the Disk Directory Mover, please read this article in conjunction with the associated article published in issue 79 of New Atari User.

DIRECT MENU ACCESS

Direct access to the menu bypassing the formatting stage presented one major problem, that of remembering the position that the directory had been moved to. Obviously, the new directory position must be entered somehow before the menu can be accessed so that the program can be set up correctly to suit a previously modified disk. Entering this via the keyboard either from ones memory or from notes could prove error prone because it is so easy to make mistakes. Some means of achieving this reliably was necessary and to this end, a disk data file was chosen.

To access the menu directly, a prompt to press the TAB key is given from the initial screen along with the directory positioning options. After the TAB key has been pressed, the data file is read and only if found on the disk currently in the drive will direct access to the menu be enabled. If the file is not found, then an error will result and OPTION must be pressed to exit denying direct menu access.

After a disk has been formatted and prepared by the program, a prompt is given to insert a standard (normal directory position) disk into the drive to receive the data file DIRMOVE.DAT which records the position that the directory has been moved to. To access the menu directly on a subsequent occasion, it is then only a matter of running the program with the disk containing the previously written data file in the drive.

The data file DIRMOVE.DAT is used to record 2 pieces of information, firstly the new directory position and secondly the format density of the modified disk for information purposes on the menu screen.

LISTING THE DIRECTORIES

The menu of the original version of the Disk Directory Mover had the options to lock/unlock files and to write with or without verify as options 7 and 8. These have been moved to options 8 and 9 respectively in version II. Option 7 is now used for directory access. Upon selecting option 7, a menu is presented from which "S" is pressed to list the directory of a standard disk and "M" is pressed to list the directory of the modified disk. But of course, the appropriate disk must be in the drive otherwise data (rubbish) will be displayed instead.

The directory is displayed in 2 columns of 16 file names per column over 2 screens displaying if necessary all 64 file names. If the directory contains 32 file names or less, then only a single screen is required and OPTION is pressed to exit. If the directory contains more than 32 file names, then a second screen is necessary and START is pressed to list the remainder of the directory on the second screen or OPTION is pressed to exit.

OTHER MAJOR CHANGES

The initial screen has been slightly altered and the VBI routine has been modified to flash the arrow used to select the new directory position and to flash the data entry cursor. The program now amends the disk's VTOC sector(s) before writing the DOS.SYS file. In the original version after pressing ESCAPE/OPTION to exit from the menu back to the initial screen, "Y" had to be pressed to confirm the request since it was not possible to re-enter the menu. Exiting is now actioned directly upon pressing ESCAPE/OPTION.

USING THE PROGRAM

A master disk should be prepared for the Disk Directory Mover as stated in the original article published in issue 79. To avoid confusion when copying files over more than one session, it is recommended that the associated disks are kept in pairs, that is the modified disk should be kept with the standard disk which contains the files to be copied. The standard disk should also contain the data file DIRMOVE.DAT written to it during the preparation of the modified disk. In this way, several modified disks could be prepared and have files copied to them in the same session without confusion.

A WORD OF WARNING!

Do NOT select option 8 from the menu to lock/unlock files if a standard directory disk is in the drive since it could corrupt the disks data. This option writes into the moved directory position to lock/unlock files which could be in data sectors of a standard disk. The Disk Directory Mover has no way of determining whether the disk in the drive is a standard disk or a modified disk so please be careful! ●

The Disk Directory Mover program can be found on the Issue Disk

USER vs PROGRAMMER

Joel Goodwin discusses the ways in which the user might see things differently to the programmer

The programmer normally enjoys a close relationship with the computer. This, of course, is essential if the programmer is to get the best out of the hardware. The user, however, is sometimes excluded from this relationship and can get well and truly left out in the cold. What might appeal to the programmer may not suit the user so well. The user looks for convenience and practicality, while the programmer looks for internal efficiency and finesse.

Above all, the computer must communicate effectively with the user and if a program is not written to achieve this then all of the work invested in it can be wasted. This article is about designing software for ordinary people who are often forgotten or ignored during the creation of a program.

WELCOME TO THE MACHINE

Just to demonstrate how important design can be, consider a few examples. Do you prefer a text menu with keyboard input or a windows system driven by a mouse? Are you enthusiastic about BASIC giving you the error message 'ERROR- 162' or would you prefer 'Disk Full'? Do you really enjoy learning CTRL key commands for utilities such as Textpro? Is a blank screen reassuring or would you prefer a 'Please Wait - Initialising' message?

The subject being discussed here has many names. A common term is 'user-friendliness'. De Re Atari has a wonderful appendix on it called 'Human Engineering'. A modern term is 'Human-Computer Interaction' (HCI). Whatever name you care to use, it is something that should always be considered when writing programs. There are, unfortunately, no hard and fast rules about the best way to design software; as De Re Atari puts it, it is 'an art, not a science'. The aim of this article is not to supply answers but to encourage programmers to ask themselves questions about what the user really wants. We'll discuss some topics and examples to illustrate the types of problems and the solutions that exist.

THE LEARNING CURVE

What is the first thing you do when you get a new piece of software? Normally, you would probably just shove in the new cassette/disk and boot it - only later if necessary would you look at the instructions. Everyone has an instinctive loathing of manuals, so a program will be more successful in a user's eyes if it is designed to minimise the importance of a manual.

Software, to some extent, should maintain independence from the manual. A program should be able to lead, if not teach, a user. If a program forces a user to read the accompanying instructions, the user will become disenchanted with the program from square one. How is this avoided? There is no single answer, but it would seem sensible to design a program which keeps the user well-informed at all times: What options are available? What exactly do they do? If an error occurs, what does it mean and how do you remedy it?

The problem with this is that the computer does not possess an infinite memory capacity in which to store bags of informative text messages. What is important is a balance. On the DOS 2.5 menu, the option 'A. DISK DIRECTORY' is sufficient. I don't need 'Press "A" to call up the disk directory, which is a list of files on the disk currently located in the specified drive'. Too much information can be as bad as too little; it can impede progress as the user cannot isolate the key points. A 'HELP' function can assist in allowing the user to control the level of information being received.

The faster the user can pick up a program, the more the user will like it. Although communicating with the user is very important it is not the only factor which can affect the learning curve associated with a program. Consider the method of requesting or inputting data in a utility program. If you keep a consistent approach to input (e.g. all input is through a cursor controlled by joystick) then the user always knows what to do. An inconsistent approach can confuse: I have seen many a program which has some menus which act on a single keypress and others which require RETURN to be pressed after the letter. It is all very well telling the user to press RETURN, but why should the user have to? A single menu subroutine can be useful here. (Note: De Re Atari advises that the keyboard should only be used as an input device if unavoidable, e.g. filename entry.)

CLARITY

Graphics can be a burden as much as a blessing. Attractive graphics may simply turn out to be clutter where the essential details are not highlighted, or worse, are obscured. Colour, symbols and animation can all enhance the display but care must be taken to avoid overuse. I recall Clayton Walnum wrote in ANALOG magazine about an occasion where he had used character set cycling to

produce pulsating orbs around the display of a text adventure. He'd slaved over the routine and graphics for some time. Once it was working, he showed it to someone (his wife, if memory serves) and he was told that it looked very nice... but it was distracting. Eventually he realised that he had to take the animation out: it was ruining a fine text adventure (Clayton did go on to write an ANALOG article called 'Textually Graphic' on the same subject as this article).

Something that is hard to come to terms with is the need to remove a perfect piece of code purely because it doesn't look right. It is worth it, believe me. An overuse of graphical variety will just look like a uniformity of chaos.

Sometimes simplicity is the best approach. A nice example is John Foskett's 'Sound Selector' (NAU issue 76) which features little more than a screen full of text. However, the program slightly modifies the display list and uses DLIs to stratify the screen and this works very well. Not only does it introduce clarity, but it brings life to what could easily have been a dull text display.

Symbols are good to use if there is some natural connection between the symbol and its meaning. For instance, a pair of scissors can be used to represent a 'cut' option. A shield icon can be used to label a bar of shield energy in a game. Whenever devising a set of symbols to represent concepts or objects, try to develop symbols which seem natural, something anyone could understand - or at least within the general context of the program. Think how you might express the following:

i) Music
ii) Time
iii) Eat
iv) Money
v) Save

Of these, (v) is probably the most difficult. 'Save' doesn't appear to have a natural symbology attached to it. Something more inventive would be required; one way of expressing it would be to have a disk with an arrow pointing to it. Note that this symbol is safely dependent on context.

The last thing I'll mention in this section is character sets. Make sure that the set you use for text is easy to read. I have a copy of 'Lemmings 2' for the Sega Mega Drive (I make no excuses for my heresy) and the text font it uses is appalling. 'C' looks like 'G'; 'M' like 'N' and 'I' like 'J'. This font is used to display passwords which are 20 letters long and cannot be found in any dictionary I know of. Make one mistake writing down the password and the whole thing is invalid. This font does not make me very happy.

THE FRUSTRATION FACTOR

Staying with 'Lemmings 2', there is a 'Nuke' option you can activate if you feel you cannot complete the level. You move the cursor onto a mushroom cloud icon and press the button twice - this is good for safety. However, you then have to wait for every single lemming to explode, each with a 5-second timer on their head. How pretty and wonderful... the first time. After that, it becomes annoying. So you get used to hitting the 'Fast Forward' icon which speeds things along. But this means every time I need to restart the level, which I need to do a lot, I have to press the button three times with a deft joypad movement in the middle to change icons - and I still have to wait a short while. When there are 50 lemmings on the screen to explode, things tend to slow down a little. Forcing this upon me every single time really puts me off playing. Many other games have the same problem: why should I have to wait for the death march to play? Or for the ambulance to come on? Or for the car to explode into many beautiful pieces bouncing around the screen? All I want to do is get on with my next go. Any programmer's idea of fun is not welcome when it strongly interferes with gameplay. Repetition in this way can kill a promising program from the outset.

An extremely common problem is the dreaded temporary text message. That is, text which appears and takes either too long to disappear or disappears too early and you missed half of it (watch the Chart Show on ITV for a demonstration of this, especially as those messages disappear from the END first). Unless it's a message of few words, my advice is always use a prompt. Never try to estimate how long the user will need to read a mass of text; just use 'PRESS SPACE BAR TO CONTINUE' or something in that vein. For the same reason, I detest long scrolling messages - I could quite easily read a large page of text which would be a lot quicker with a lot less eyestrain. Yes, I am the Victor Meldrew of the computer world.

These are indirect examples of communicating too much to the user, or at least too slowly. Problems of frustration are not always of this type, though. Most of this article has revolved about the communication of information between user and computer. Games, especially, suffer from problems unrelated to information per se. There are many games, for example, which need a character to be positioned in EXACTLY the right place to jump a hole and not crash into something on the other side. One pixel off and I'm dead. This is not good gameplay. This is sadism.

Let me describe to you a problem I have had to deal with and how I resolved it. When I wrote the Citadel (NAU issue 74 disk bonus) I faced the problem of what speed to give the player's character. The character jumps from position to position; as one jump too many can ruin the player's progress, judging the correct time delay between moves/jumps was important. However, a puzzle game like the Citadel attracts a variety of players; some will have arcade game backgrounds adept at joystick use, others will be into more cerebral games which are not demanding in terms of joystick reactions. The former will like a fast speed, the latter a slow speed. So a single game speed would not suffice - it was clear that I had to supply a speed option on the title screen.

Orson (NAU issue 71) faced a similar problem but it is not so pronounced. In Orsòn, the character can be seen to walk from one position to the next and doesn't jump. This allows the player to gauge when to release the joystick and a speed setting is not necessary. However, the story does not end here. I have since realised what the ideal solution is which avoids any effort on the part of the player - a joystick interface which prevents continuous movement. That is, if the joystick is pushed left then the character would move left ONCE. The player needs to return the stick to neutral then push left again to move the character another step. This totally eradicates the movement problem: the character automatically moves at whatever speed the player is capable of. This solution requires more work than both the Citadel and Orson implementations but it would be better by far.

CLOSING NUMBER

There is an awful lot to consider when designing a program. I've tried highlighting some areas, even discussed some specific problems, but this subject is just too large to

condense into a small article. Let me re-iterate a few important ideas:

- Put yourself in the user's shoes. If this involves extensive testing by yourself or a friend (soon to be ex-) then so be it!
- The user learns quickly when a program behaves in a consistent way across a variety of circumstances.
- Don't be afraid to remove portions of code because they don't work well with the user. Your program will be better for its 'lobotomisation'.
- Try to talk in a language the user understands whether by symbols or text.

It is instructive to look at other people's programs and think about what annoyed you about them. Learn from this; don't make the same mistakes yourself. It can also be useful to consider what you liked best about a program.

The subject of HCI is receiving a lot of attention these days and many books have been written about it. There is a vast range of theories and practical approaches concerning the design of the interface with the user and I have found a little reading on the side has helped a lot. The Open University book 'Human-Computer Interaction' (Jenny Preece et al., ISBN 0-201-62769-8) provides an excellent comprehensive introduction to modern HCI problems and solutions; if not wholly applicable to the Atari, it will certainly get you thinking.

I hope I've provided some food for thought. If anyone disagrees with anything presented in this article then drop a line to Mailbag - I'm sure they'll be pleased to print your comments. Good luck on your next program! ●

SUPERCIRCLES
continued

needed. Also, the subroutine could probably be extended to become a "superellipse" subroutine.

Perhaps this isn't an "optimum" method and someone else has a better idea, which relies only on integers perhaps. If so, please write to NAU, I'd love to hear of it!

MATHEMATICS!

The equation of a circle about the point (0,0) with radius r is given by:

$$x^2 + y^2 = r^2$$

Differentiating this equation with respect to x yields:

$$2x + 2yy' = 0$$

We can approximate the derivative term y' by using small changes in x and y, denoted by Dx and Dy. So this equation can be approximated by:

$$2x + 2y \frac{Dy}{Dx} = 0$$

Re-arranging:

$$Dy = -\frac{x}{y} Dx$$

So if we move along the circle by Dx in the x-direction, then we need to move by Dy, specified by this last equation, in the y-direction. If we restrict ourselves to Dx=1, then this tells us exactly where to put the next point on the screen display. This will not work if you have the correct next point vertically adjacent; if we increase with Dx=1 we will only see one point and then advance to the next column. We should now switch to using Dy=1 and determining the corresponding Dy. Fortunately, by using symmetry we stop as we reach such a point. ●

PD ZONE

by Austin Hillman

We return to normal service this issue with yet another random selection taken from the public domain.

READ ALL ABOUT IT

Codesmiths Newsletter Reader (DS#48) by Sean Puckett is a program that will allow you to produce a classy looking disk based newsletter. The CNR reads standard text files into which commands are inserted to control the features of this program. The CNR automatically word-wraps and microspace justifies the text for you.

The commands available are as follows.

Load font - Loads a font into one of the two font buffers available. Proportionality information is derived from the font data itself.

Select Font - Selects either one of the two fonts for printing by the program.

Wait for a Key - This pauses printing until you press an key. Whenever the computer is waiting for a key, the border around the screen becomes a different colour than the background.

Page Break - Halts the printing, and waits for a key. CNR will wait for a key if the screen is about to scroll, so this is useful only as a chapter break or something like that.

Clear Screen - Instantly clears the screen and starts printing from the top.

Set screen colours - The first parameter sets one of sixteen possible hues for the screen colour. The second parameter is the intensity of the screen. The third parameter is the brightness of the text itself.

Load Screen File - This clears the text screen, sets a graphic mode, and loads 7680 bytes of image data directly so the screen. The four colour registers are stored as the last four bytes of the file in Micropainter format. After loading the image, the CNR waits for a key, then returns to a clear text screen and continues printing.

Global Borders and Line Spacing - Sets three important global settings. Left margin, right margin and line spacing.

Menu Option Definition - Sets the option string referred to by the number parameter to the filename parameter. This, when combined with menu selection makes this a powerful program.

Menu selection - This command waits for a number key to be pressed. The number must be greater that or equal to low, and less than or equal to high. The number pressed is stored internally. X when used in place of a filename means use the filename we stored with the menu option definition command, instead of an X.

Execute sub-file - Reads and prints and executes a subroutine-like file.

Top of file - This re-executes the current file.

Chain to file - Similar to execute sub-file, except that the current file is ended, and the

new file replaces it.
If statement - Skips the next line if the number key pressed was NOT the parameter.
End file - This is not required at the end of a file, it just serves for a useful way to exit a subroutine or loop when you test with an if command.
Define a label - This marks a position in the file that you will be able to jump to with the next command.
Jump to label - This causes the CNF to skip all instructions and text until a label is found that matches the jump command. Note this will only jump forward.
Make sure file exists - This command will check for the existence of a file (or X). If the file does not exist, the CNR waits for a key, then re-checks for the file. This command will not allow the CNR to continue until the file it wants is found. Use this command to verify that the user has switched disks if you asked him to.

The example newsletter provided is a tutorial on the commands listed above and how best to use them. Several digitised picture files are also included.

Even thought his appears to be an excellent piece of software I have not yet come across any disk magazine that uses the program, which seems a pity.

GOOD GAME

Ultra Tetris (#190) is a Turbo-BASIC version of the popular falling bricks game written by Tim Truesdale. In this version the pieces are rectangular and composed of three coloured squares. The object is to get three or more squares of the same colour into horizontal, vertical or diagonal lines.

The first five levels are selectable, but you have to beat the next five by skill alone. Control is by keyboard for left, right, rotate, drop and pause. A high score table is provided, as is a good looking title screen. Released on a shareware basis, the registered version has instructions on how to unlock the hidden features of this game. If you like Tetris I think you will find this a good variation of the original.

HELPING HAND

Extended Atari BASIC (#186) by Robert Berry adds extra commands and features to standard Atari Basic. It is suitable for the XL/XE only as it utilises the shadow ram in these machines and thus only uses 3k of free ram.

On loading the standard screen appears with an extra 25th line at the top telling you EAB is installed. Also on this line a clock counts the elapsed time until it is set to the current time by the appropriate command. The extra line will remain active until a graphics call is made.

The program is disabled when DOS is called and so adds 10 DOS commands to reduce the need to do so. They are DIRectory, LOCK, UNLOCK, DELETE, RENAME, Write DOS, FORMAT, SINGLE density format, BSAVE and BLOAD. Another 11 commands help with Basic programming including a PROTECT command which will 'lock up' the computer to keep it safe from interference. As well as AUTOnumber, RENumber, Variable Name Table, VERIFY, DEC, HEX, SIZE, LOMEM, ERROR and CONVERT commands.

Special function commands control the cassette, printer and 130XE rambank as well as providing coldstart and Basic off from the keyboard. Also available is a joystick controlled cursor, flashing if you prefer, 5 predefined and 7 user defined keys for your convenience.

Most of the extra commands may be utilised by a Basic program via an XIO call. Sample programs are provided to demonstrate the CONVERT, FLASH, ERROR and BANK commands. Full documentation of all commands and features is provided.

Extended Atari BASIC is a neat implementation of many useful commands and utilities for the Basic programmer. That being said, Turbo-BASIC has now superseded this program and is clearly the first choice for serious programmers.

A NEW LOOK FOR OLD FONTS

Glyph Font Editor (#212) written in Action! by Jack Prevost is a fully featured font editor. The layout is clean and simple with a font display, edit window and menu table. The menu has 21 commands which are selected by arrow key or joystick, the edit window is also controlled by either method. Two new fonts plus the system font can be held in memory at once and can be swapped at will. The system font is for reference only.

Creating a new font or altering an existing one is simplicity itself, but if you are feeling lazy, nine new fonts are included to show you what can be achieved with this program.

Some handy utilities have also been provided along with full instructions. FONTLOAD.COM allows SpartaDOS users to load a non resident font from the command line. ROSETTA.COM converts any font into BASIC data tables or an Action! data block for use in your programs with the FONT.LST or FONT.ACT loader programs supplied.

WRITE ON

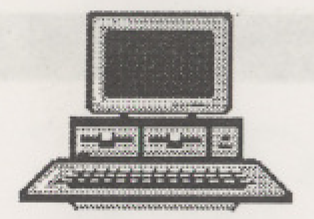
Chick Scratch (#147) by Robert Chick is a word processor for the XL/XE. On bootup you are presented with a red screen and a flashing cursor with a command line at the bottom. Editing commands are the usual CONTROL letter or OPTION plus CONTROL letter combinations. Print format commands are SELECT letter combinations. They are set up to suit the XMM801 printer but can be changed to suit any other printer. Commands are summarised on five help screens available by pressing CONTROL H. The usual miniDOS screen completes the package.

Needless to say if the default settings are not to your liking they may be altered. The customising program provided allows you to permanently change the default settings for text/screen colour, upper/lower case, insert/overwrite mode, screen width, print format, key click and text/program mode.

The full documentation supplied provides all you need to know. It closely resembles other word processors like Speedscript and is thus easy to use if you are already familiar with the layout, but even for the beginner it is a good choice. It may lack some of the features that more expensive word processors have, but it is a very competent program as I have discovered while using it to produce this article.

RATINGS

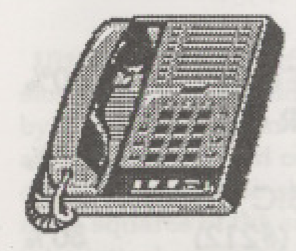
CODESMITHS NEWSLETTER READER (DS#48)	60%
ULTRA TETRIS (#190)	70%
EXTENDED ATARI BASIC (#186)	50%
GLYPH FONT EDITOR (#212)	80%
CHICK SCRATCH (#147)	90%



JOURNEY INTO CYBERSPACE

TO AMERICA VIA THE INTERNET

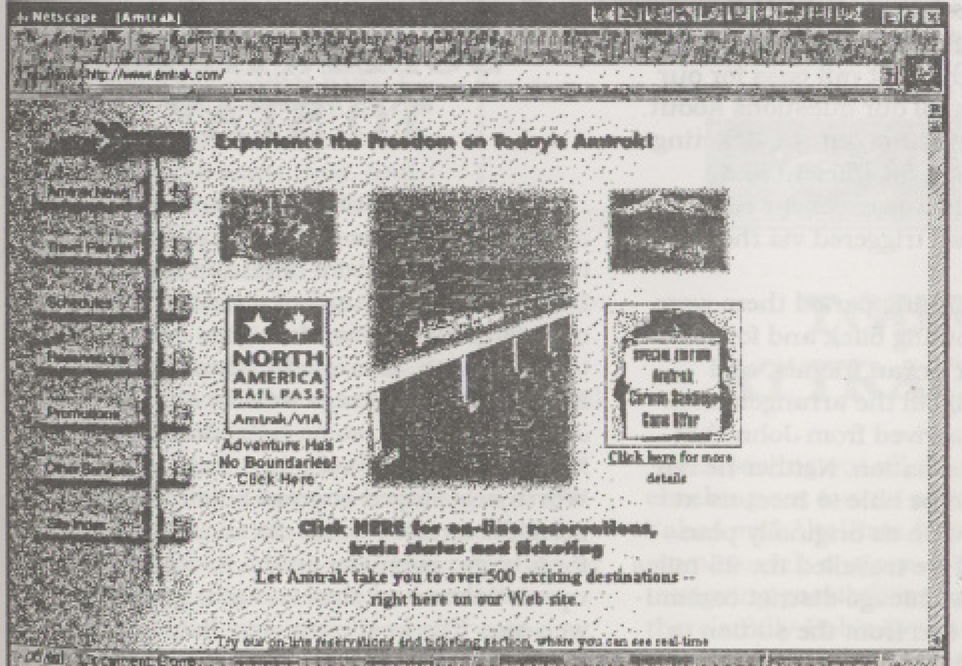
John S Davison continues his exploration of the Internet



In the last issue we looked at how my son John jnr. used the Internet to help arrange his emigration to the USA. Several months after his departure we decided to visit him at his new home near Chicago. Also, we have friends in Fort Worth, Texas we'd not seen for several years, so we decided to include them on our itinerary. Following John's success with the Internet we thought we'd use it to plan our trip too.

The major expense of any trip to the USA is the air fare, but there are lots of on-line places you can look to find special offers on flights (and complete holiday packages if you need them) to help keep the cost down. For instance, if you're a Compuserve member the command GO HOLIDAYS will take you to an area that claims to have 70,000 holiday and flight deals on offer. We have an old friend who works in the travel industry who can get us really good deals on flights to the USA, but unfortunately he had no tickets to Chicago available in the timeframe we wanted. The best flights he could offer were from Gatwick to Houston, Texas, so we decided we'd start our trip in Texas. A quick scan of prices of similar flights on the Internet and Compuserve showed our friend's offer beat the on-line deals, so we booked through him.

One small problem - Houston is about 250 miles from where we really wanted to be - the Dallas/Fort Worth (DFW) area - so we needed an internal flight to take us there. A quick visit to Yahoo (www.yahoo.com) pointed us at SouthWest Airlines (www.iflyswa.com) who were advertising special promotional deals on flights within Texas. The fare to DFW was very reasonable and I could have booked it on-line there and then. However, I can't yet bring myself to trust my credit card details to the Internet, so decided to book through a travel agent instead. Unfortunately, we then discovered that SouthWest Airlines don't subscribe to the booking systems used by UK travel agents, so we couldn't book directly. The agent checked what else was available, and found that American Airlines were also doing special promotional deals on Texas routes and could actually substantially undercut SouthWest's low ticket prices, so we booked with them instead. The total cost of the two flights to get us to DFW was less than any direct flight I could find, so we were doing OK so far.



PLANE OR TRAIN?

It's around 1100 miles from Fort Worth to Chicago, and the sensible way to travel between them is obviously by air. So did we do the sensible thing and book these flights too? Oh, no! We had this crazy idea of going by train instead. The thought of travelling across Texas, Arkansas, Missouri, and Illinois using a mode of transport unknown to most Americans was appealing, as it added an extra element of adventure to the trip. So, it was back to Yahoo again to find details of train companies. We ended up at Amtrak's web-site (www.amtrak.com), the company that now handles virtually all long distance rail travel in the USA.

The site described all Amtrak's long distance routes, and to our delight we discovered they offer special USA visitors' rail passes. These were just what we wanted - unlimited travel by train for 15 or 30 days in various zones of the USA. The one which would get us from Fort Worth to Chicago and back again cost the equivalent of about £130, a real bargain. This covered "coach class" seats only, with basic or de-luxe sleeper cabin accommodation available at (considerable) extra cost. Travel would be on the "Texas Eagle", a train that runs from San Antonio in southwest Texas via Fort Worth all the way up to Chicago. The web-site also



All the information you will need to ride

Metra's Commuter Rail System

Serving Northeast Illinois

Copyright 1998, Northeast Illinois Regional Commuter Railroad Corporation

General Information

Americans With Disabilities Information

showed the train's timetable, so we could work out our other travel plans around it. These long distance trains only run every two or three days, so you have plan accordingly. If you miss one it could wreck the rest of your schedule.

Further details were available by filling in your name, address, and phone number in an on-line form, so we did this. A few days later we were surprised to receive a phone call (yes, they phoned us!) from Leisure Rail, AmTrak's UK agent. AmTrak had informed them that we were interested in USA rail passes, so were calling to see how they could help. In fact they were extremely helpful over the following few weeks, advising on the best rail pass for our purposes, answering all our questions about USA rail travel, and sorting out the ticketing and payment details in an efficient and friendly manner. This was excellent customer service - and it was all triggered via the Internet.

Throughout the planning period there were e-mail messages zooming back and forth between John jnr., our Texan friends, and ourselves, confirming all the arrangements. One day a message arrived from John that caused a little consternation. Neither he nor his wife Alison would be able to meet us at Chicago's Union Station as originally planned. They suggested we travelled the 25 miles out to Wheaton on a Chicago district commuter train and take a taxi from the station to their apartment and they'd meet us there. So, it was back to Yahoo again where we found a web-site containing route maps and the complete current timetable for Metra Rail (www.metrarail.com), Chicago's local train system! Details of the Chicago-Wheaton service were duly added to our growing collection of travel documentation.

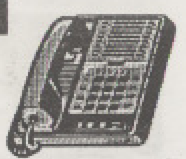
SOMEWHERE TO STAY

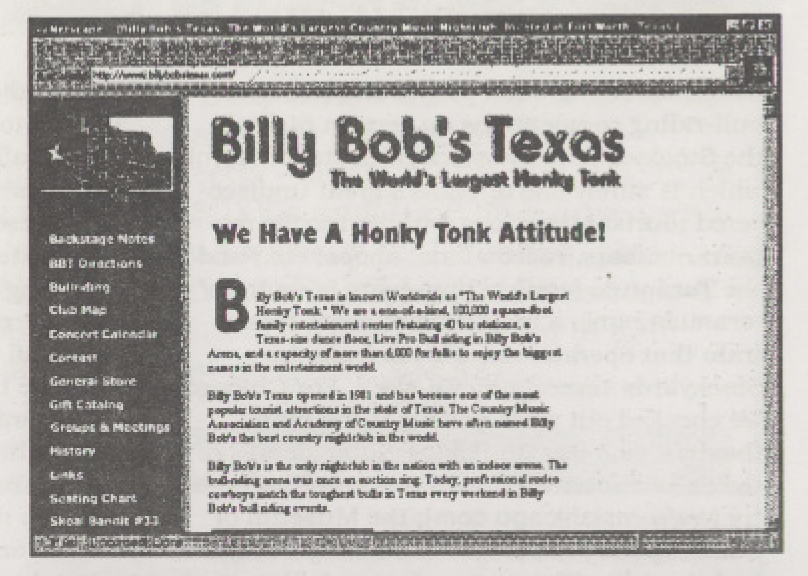
Accommodation was the next area to tackle, to cover our now somewhat bizarre looking itinerary. We'd need a hotel somewhere in the DFW area for a couple of nights at the start of our visit, another for six nights on our return to Fort Worth, and one in Houston for our final night. In Chicago we'd be staying with John and Alison, so no hotel needed there. Many years ago I had a business trip to Fort Worth and stayed at great hotel, the Worthington, for a ridiculously cheap price (about £50 for a large, executive suite). So we looked it up on Yahoo and sure enough, it has its own web-site. The best room deal listed there was now a wallet-busting £160 per night, so we decided to check out Holiday Inn prices instead. There are Holiday Inns all around the DFW area, so via their web-site (www.holiday-inn.com) we priced several for comparison purposes and eventually settled on one that suited our budget.

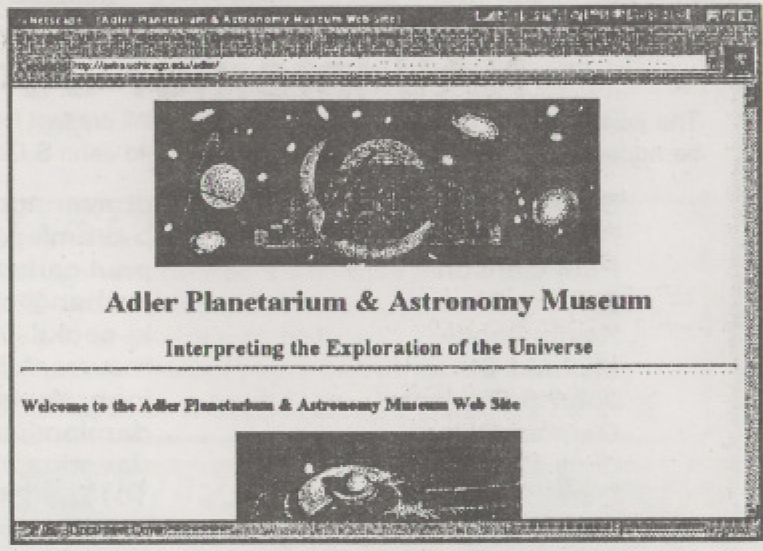
You can also check availability of Holiday Inn rooms on-line for the dates and type of room you want. To our chagrin we discovered that our first choice hotel was fully booked. Number two on our list had rooms available, so we went for that. Again we could have booked on-line, but the credit card number paranoia set in once more and we called the free Holiday Inn reservation number instead. The person taking the booking tried to charge us a higher room rate than we were expecting, but relented when I quoted the rate we'd seen on the web-site. We booked the hotel for Houston at the same time, and got the web-site rate for that too.

We decided to wait and see how much cash we had left at the end of our Chicago stay before booking accommodation for the return visit to Fort Worth. In fact, our Texan friends found a really good deal for us, and this was sorted out via e-mail before we left Chicago.

The only other essential to take care of was the car hire, but as we were staying near the airport we figured we could get a good deal locally when we arrived, and this proved to be the case. But you can look up rental rates and/or book on-line with the major companies if you wish.







TOURIST ATTRACTIONS

After sorting out all the essentials we started checking out tourist attractions, using the Yahoo and AltaVista (www.altavista.digital.com) search engines. We found dozens of intriguing possibilities here - in Fort Worth there's Billy Bob's (www.billybobstexas.com), the biggest bar and night-club in the world



(accommodating 6000 people and with live bull-riding competitions on certain nights!); the Stockyards (www.stockyardsstation.com), which is surely one of Texas's great undiscovered tourist attractions with its terrific restaurants, bars, rodeo arena, shops; etc.; and the Tarantula (www.ci.grapevine.tx.us/train/tarantula.html), a restored wild-west steam train that operates excursions from the Stockyards several times a week. For Chicago we checked out what was on at the many theatres, and jazz and blues clubs; details of the excellent Museum of Science and Industry (www.msichicago.com); the Museum of Contemporary Art (www.mcachicago.com); and the Adler Planetarium (astro.uchicago.edu/adler).

Thanks to the Internet we were able to do virtually all the research for this trip from home. If we wished we could also have made on-line reservations and payment for air travel, hotels, and car hire. During the trip, everything turned out pretty well as our Internet research had shown - even the Chicago Metra Rail commuter train we caught departed at the time stated on the Internet timetable. Overall, our experience backed up John Jnr's. findings - that the Internet really is a usable, practical tool for this sort of activity. So, if you need to plan a trip, give it a try - you'll be amazed at the useful facilities and material out there. ●

NAU Internet Contact List

The following NAU readers would welcome e-mail contact from other Atari users. If you'd like to be added to this list please drop an e-mail note to John S Davison at the address below.

Daniel Baverstock	dbaverstock@mistral.co.uk
Paul Bramley	p.bramley@student.qut.edu.au
Paul Carlson	paul.carlson@hn.se
Johnny Chan	jwchan@clara.net
Kevin Cooke	kj-cooke@wpg.uwe.ac.uk
Michael Current	mcurrent@carleton.edu
John S Davison	john_davison@compuserve.com
Damian Dixon	damian@tenet.co.uk
Gary Dundas	davadar@hotkey.net.au
Derek Fern	101755.2443@compuserve.com
Dean Garraghty	dgs@clara.net
Joel Goodwin	jgoodwin@wilco.co.uk
Paul Herbert	147378.97@swansea.ac.uk
Gordon Hooper	ua558@freenet.victoria.bc.ca
Fred Meijer	fmeijer@dsv.nl
Ann O'Driscoll	annod@iol.ie
Allan Palmer	Allan_Palmer&bigfoot.com
Paul Rixon	rixonp.railtrack@ems.rail.co.uk
Paulo A Rodrigues	nop25450@mail.telepac.pt
Brad Rogers	brad@pianosa.demon.co.uk
Nigel Turton	npturton@msn.com
Henning Wright	kofta@algonet.se
Daniel Yelland	yh182530@stmail.staffs.ac.uk
Bryan Zillwood	b.j.zillwood@exeter.ac.uk

contact ... contact ... contact ...

FOR SALE

800/400 SYSTEMS: 800XL computer, 1050 disk drive, 1010 program recorder, 400 'Maplins' 48K upgrade, 410 program recorder, power supplies, leads, manuals, joysticks/paddles, ROMs, disks, tapes, magazines 1981 to present day. Excellent condition - in storage for several years. £100. No splits, no offers, buyer collects. Contact S. Campbell on 01753 536707 (Slough, Berks)

LOTS OF STUFF: Atari 8-bit hardware & software. For list please contact Mike on (01302) 834410 or e-mail hamster@global-net.co.uk

FOR SALE

SURPLUS HARDWARE: 8-bit hardware for sale, surplus to needs - 800's, 800 & 600XL's, 65 & 130XE's and 1050 disk drives with and without US Doublers, plus other odds and ends. Please telephone Richard for an up to date list on 01202 256927

WANTED

ROM CARTRIDGES: ROMs for Atari XL/XE wanted, boxed with instructions preferred. Also original copy of LEADERBOARD for Atari ST/E. Call Chris on 0049 2163 990329 (Germany)

FREE TO SUBSCRIBERS

The CONTACT column is free of charge to subscribers who wish to sell their equipment or contact other readers. Space is limited so we request that entries be kept as short as possible. Extremely long entries may be heavily edited or ignored. Send your CONTACT notice on a separate sheet of paper (not as part of a letter) to:

CONTACT, PAGE 6 PUBLISHING, P.O. BOX 54, STAFFORD, ST16 1DR

FOR SALE ... WANTED ... PEN PALS ... ADVICE ... HELP

PROGRAM LISTINGS

Certain program listings which are too long to include in the magazine may be obtained free of charge as printed listings to type in. All programs are, however, included on the Issue Disk which is available with each issue. Remember this disk also includes BONUS PROGRAMS which do not appear in the magazine. If you would like the type-in listings please write or telephone indicating which listings you require. Please note that there are not necessarily extra listings for every magazine.

Write to LISTINGS, NEW ATARI USER, P.O. BOX 54, STAFFORD, ST16 1DR or telephone 01785 241153